Sparse Linear Models

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

April 10, 2019

Contents

Following closely Chapter 13, of K. Murphy, Machine Learning: A probabilistic Perspective

Introduction

Sparse Linear Models: Introduction

- We focus on selecting sets of variables at a time using a model-based approach.

- We consider each variable in the presence of the rest.

- If the model is a generalized linear model, of the form $p(y|\boldsymbol{x}) = p\big(y|f(\boldsymbol{w}^T\boldsymbol{x})\big)$ for some link function f, then we can perform feature selection by encouraging the weight vector to be sparse (have lots of zeros).

Sparse Linear Models: Introduction

- Some applications where feature selection/sparsity is useful

  - In many problems, $D \gg N$. The corresponding design matrix is short and fat, rather than tall and skinny. This is called the small N large D problem.

    We want to find the smallest set of features that can accurately predict the response.

  - Sometimes, we use basis functions centered on the training examples, $\phi(\boldsymbol{x}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_N)]$. Feature selection in this context is equivalent to selecting a subset of the training examples, which can help reduce overfitting and computational cost.

Sparse Linear Models: Introduction

- In signal processing, it is common to represent signals (images, speech, etc.) in terms of wavelet basis functions. To save time and space, it is useful to find a sparse representation of the signals, in terms of a small number of such basis functions. This allows us to estimate signals from a small number of measurements

- Feature selection and sparsity is currently one of the most active areas of machine learning.

- Balding, D. (2006). A tutorial on statistical methods for population association studies. *Nature Reviews Genetics 7*, 81–91.

Bayesian Variable Selection

- Let,

$$\gamma_j = \begin{cases} 1, & \text{if feature } j \text{ is relevant} \\ 0, & \text{otherwise} \end{cases}$$

- Our goal is to compute the posterior over models

$$p(\gamma|\mathcal{D}) = \frac{\exp\left(-f(\gamma)\right)}{\sum_{\gamma'} \exp\left(-f(\gamma')\right)}$$

where $f(\gamma)$ is the cost function

$$f(\gamma) \triangleq -[\log p(\mathcal{D}|\gamma) + \log p(\gamma)]$$

- For example, suppose we generate $N = 20$ samples from $D = 10$ dimensional linear regression model, $y_i \sim \mathcal{N}(\boldsymbol{w}^T \boldsymbol{x}_i, \sigma^2)$ in which $\mathrm{K} = 5$ elements are non-zero.

- In particular, we use

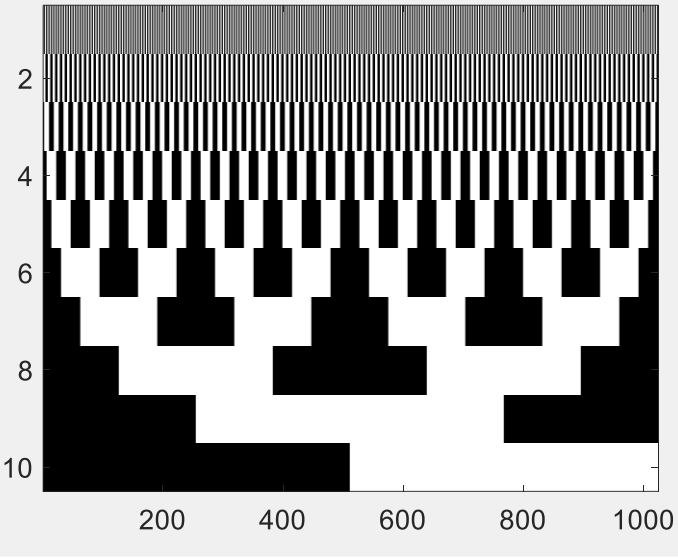
$$\boldsymbol{w} = \{0.00, -1.67, 0.13, 0.00, 0.00, 1.19, 0.00, -0.04, 0.33, 0.00\} \text{ and } \sigma^2 = 1$$

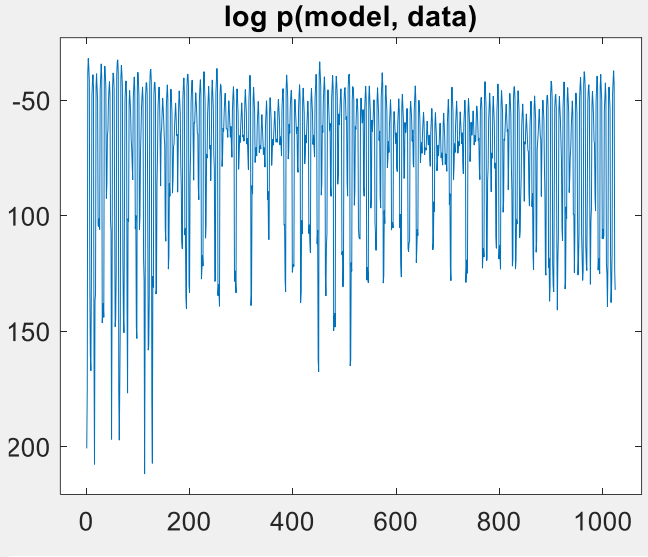
Bayesian Variable Selection

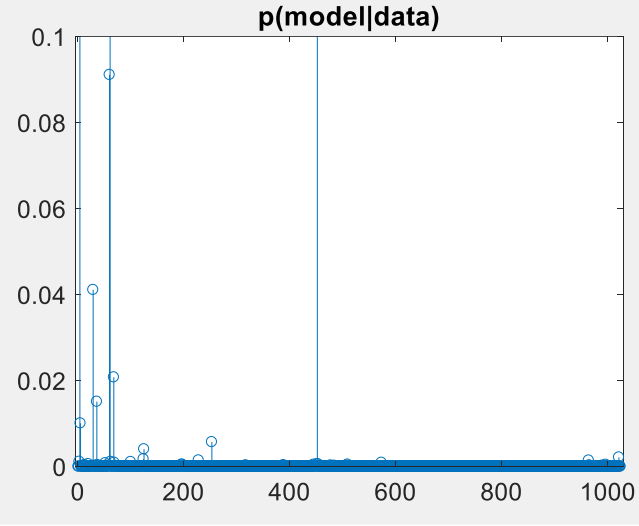
- We enumerate all $2^{10} = 1024$ models and compute $p(\gamma|\mathcal{D})$ for each one.
- We order the models in Gray code order, which ensures consecutive vectors differ by exactly 1 bit.
- We will see that the objective function $\log p(model, data)$ is extremely bumpy.
- The results are easier to interpret if we compute the posterior distribution over models, $p(\gamma|\mathcal{D})$.
- The top 8 models are listed below.
- The "true" model is $\{2,3,6,8,9\}$.

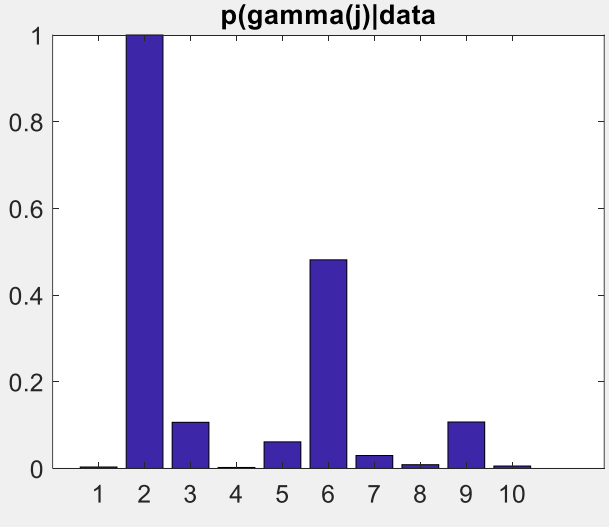
model	prob	members
4	0.447	2,
61	0.241	2, 6,
452	0.103	2, 6, 9,
60	0.091	2, 3, 6,
29	0.041	2, 5,
68	0.021	2, 6, 7,
36	0.015	2, 5, 6,
5	0.010	2, 3,

Bayesian Variable Selection









(a) All possible bit vectors of length 10 enumerated in Gray code order.
(b) Score function for all possible models.
(c) Posterior over all 1024 models. Vertical scale has been truncated at 0.1 for clarity.
(d) Marginal inclusion probabilities.

linregAllsubsetsGraycodeDemo.m
from PMTK3

Bayesian Variable Selection

- Detecting the true model for the problem discussed before is extremely difficult as the features associated with 3 and 8 are extremely small compared to σ^2.
- Given enough data, the method will converge on the true model (assuming the data is generated from a linear model), but for finite data sets, there will usually be considerable posterior uncertainty.
- Interpreting the posterior over a large number of models is quite difficult, so we will seek various summary statistics. A natural one is the posterior mode, or MAP estimate

$$\hat{\gamma} = \arg\max p(\gamma|\mathcal{D}) = \arg\min f(\gamma)$$

- A better summary is the median model.

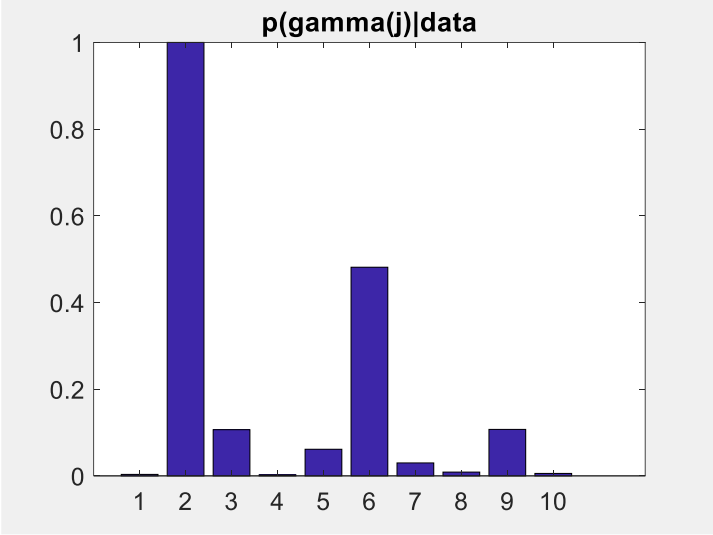
$$\hat{\gamma} = \{j : p(\gamma_j = 1|\mathcal{D}) > 0.5\}$$

- Barbieri, M. and J. Berger (2004). Optimal predictive model selection. *Annals of Statistics 32*, 870–897.
- Carvahlo, L. and C. Lawrence (2007). Centroid estimation in discrete high-dimensional spaces with applications in biology. *Proc. Of the National Academy of Science, USA 105*(4).

Bayesian Variable Selection

$$\hat{\gamma} = \{j : p(\gamma_j = 1|\mathcal{D}) > 0.5\}$$

- This requires computing the posterior marginal inclusion probabilities, $p(\gamma_j = 1|\mathcal{D})$.
- We see that the model is confident that variables 2 and 6 are included. If we lower the decision threshold to 0.1, we would add 3 and 9 as well.
- However, if we wanted to "capture" variable 8, we would incur two false positives 5 and 7.



- Variable selection is most useful in the cases where the number of dimensions is large.
- Since there are 2^D possible models, it will be impossible to compute the full posterior in general, and even finding the MAP estimate or marginal inclusion probabilities will be intractable.

The Spike and Slab Model

The Spike and Slab Model

- The posterior is given by

$$p(\gamma|\mathcal{D}) \propto p(\gamma)p(\mathcal{D}|\gamma)$$

- It is common to use the following prior on the bit vector

$$p(\gamma) = \prod_{j=1}^{D} \mathcal{B}er\left(\gamma_j|\pi_0\right) = \pi_0^{\|\gamma\|_0}(1-\pi_0)^{D-\|\gamma\|_0}$$

where π_0 is the probability a feature is relevant, and $\|\gamma\|_0 = \sum_{j=1}^{D}\gamma_j$ is the ℓ_0 pseudo-norm that is, the number of non-zero elements of the vector.

- It is useful to write the log prior as follows

$$\begin{aligned}\log p(\gamma|\pi_0) &= \|\gamma\|_0 \log \pi_0 + (D - \|\gamma\|_0)\log(1-\pi_0) \\ &= \|\gamma\|_0(\log \pi_0 - \log(1-\pi_0)) + \text{const} = -\lambda\|\gamma\|_0\end{aligned}$$

where $\lambda \triangleq \log\frac{1-\pi_0}{\pi_0}$

The Spike and Slab Model

- We can write the likelihood as follows

$$p(\mathcal{D}|\gamma) = \int \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \gamma) p(\boldsymbol{w}|\gamma, \sigma^2) p(\sigma^2) d\boldsymbol{w}\, d\sigma^2$$

- For notational simplicity, we have assumed the response is centered, so we can ignore any offset term μ_0.
- Regarding the prior $p(\boldsymbol{w}|\gamma, \sigma^2)$
  - If $\gamma_j = 0$, feature j is irrelevant, so we expect $w_j = 0$.
  - If $\gamma_j = 1$, we expect w_j to be nonzero.
  - If we standardize the inputs, a reasonable prior is $\mathcal{N}(0, \sigma^2 \sigma_w^2)$, where σ_w^2 controls how big we expect the coefficients associated with the relevant variables to be (which is scaled by overall noise level σ^2). As $\sigma_w^2 \to \infty, p(\boldsymbol{w}|\gamma = 1, \sigma^2 \sigma_w^2)$ becomes a uniform distribution (slab).

$$p(\boldsymbol{w}|\gamma, \sigma^2) = \begin{cases} \delta_0(w_j). & \text{if } \gamma_j = 0 \\ \mathcal{N}(w_j|0, \sigma^2 \sigma_w^2). & \text{if } \gamma_j = 1 \end{cases}$$

- Mitchell, T. and J. Beauchamp (1988). Bayesian Variable Selection in Linear Regression. *J. of the Am. Stat. Assoc. 83*, 1023–1036.

The Spike and Slab Model

- We can drop the coefficients w_j for which $w_j = 0$ from the model.
- Therefore can write in a compact notation,

$$p(\mathcal{D}|\gamma) = \int \int \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{X}_\gamma \boldsymbol{w}_\gamma, \sigma^2 \mathbf{I}_N\right) \mathcal{N}\left(\boldsymbol{w}_\gamma | \mathbf{0}_{D_\gamma}, \sigma^2 \sigma_w^2 \mathbf{I}_{D_\gamma}\right) p(\sigma^2) d\boldsymbol{w}_\gamma \, d\sigma^2$$

where $D_\gamma = \|\gamma\|_0$ is the number of non-zero elements in γ.

- We can generalize this by defining a prior of the form

$p(\boldsymbol{w}|\gamma, \sigma^2) = \mathcal{N}\left(\boldsymbol{w}_\gamma | \mathbf{0}_D, \sigma^2 \boldsymbol{\Sigma}_\gamma\right)$ for any positive definite matrix $\boldsymbol{\Sigma}_\gamma$.

- We can write down the marginal likelihood as

$$p(\mathcal{D}|\gamma, \sigma^2) = \int \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{X}_\gamma \boldsymbol{w}_\gamma, \sigma^2 \mathbf{I}_N\right) \mathcal{N}\left(\boldsymbol{w}_\gamma | \mathbf{0}, \sigma^2 \boldsymbol{\Sigma}_\gamma\right) d\boldsymbol{w}_\gamma = \mathcal{N}\left(0, \mathbf{C}_\gamma\right)$$

$$\mathbf{C}_\gamma \triangleq \sigma^2 \mathbf{X}_\gamma \boldsymbol{\Sigma}_\gamma \mathbf{X}_\gamma^T + \sigma^2 \mathbf{I}_N$$

- Minka, T. (2000b). Bayesian linear regression. Technical report, MIT.
- George, E. and D. Foster (2000). Calibration and empirical bayes variable selection. *Biometrika 87* (4), 731–747.
- Liang, F., R. Paulo, G. Molina, M. Clyde, and J. Berger (2008). Mixtures of g-priors for Bayesian Variable Selection. *J. of the Am. Stat. Assoc. 103*(481), 410–423.

The Spike and Slab Model

- If the noise is unknown, we can put a prior on it and integrate it out
- It is common to use $p(\sigma^2) = \mathcal{IG}(\sigma^2 | a_\sigma, b_\sigma)$.
- Some guidelines on setting a_σ, b_σ can be found in (Kohn et al. 2001).
- If we use $a_\sigma = b_\sigma = 0$, we recover Jeffrey's prior, $p(\sigma^2) \propto \sigma^{-2}$
- When we integrate out the noise, we obtain

$$p(\mathcal{D}|\gamma) = \int \int p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}, \gamma) p(\boldsymbol{w}|\gamma, \sigma^2) p(\sigma^2) dw \, d\sigma^2$$

$$\propto \left|\mathbf{X}_\gamma^T \mathbf{X}_\gamma + \boldsymbol{\Sigma}_\gamma^{-1}\right|^{-\frac{1}{2}} \left|\boldsymbol{\Sigma}_\gamma^{-1}\right|^{-\frac{1}{2}} \left(2b_\sigma + S(\gamma)\right)^{-\frac{(2\sigma_a + N - 1)}{2}}$$

where

$$S(\gamma) \triangleq \boldsymbol{y}^T \boldsymbol{y} - \boldsymbol{y}^T \mathbf{X}_\gamma \left(\mathbf{X}_\gamma^T \mathbf{X}_\gamma + \boldsymbol{\Sigma}_\gamma^{-1}\right)^{-1} \mathbf{X}_\gamma^T \boldsymbol{y}$$

- Kohn, R., M. Smith, and D. Chan (2001). Nonparametric regression using linear combinations of basis functions. *Statistical Computing 11*, 313–322.
- Brown, P., M. Vannucci, and T. Fearn (1998). Multivariate Bayesian variable selection and prediction. *J. of the Royal Statistical Society B 60*(3),627–641.

The Spike And Slab Model

- When the marginal likelihood cannot be computed in closed form (e.g., if we are using logistic regression or a nonlinear model), we can approximate it using BIC

$$\log p(\mathcal{D}|\gamma) \approx \log p\left(\boldsymbol{y}|\mathbf{X}, \widehat{\boldsymbol{w}}_\gamma, \hat{\sigma}^2\right) - \frac{\|\gamma\|_0}{2}\log N$$

where $\widehat{\boldsymbol{w}}_\gamma$ is the MLE or MAP estimate based on $\mathbf{X}_\gamma$.

- $\|\gamma\|_0$ is the "degrees of freedom".

- Adding the log prior, the overall objective becomes

$$\log p(\gamma|\mathcal{D}) \approx \log p\left(\boldsymbol{y}|\mathbf{X}, \widehat{\boldsymbol{w}}_\gamma, \hat{\sigma}^2\right) - \frac{\|\gamma\|_0}{2}\log N - \lambda\|\gamma\|_0 + \text{const}$$

- We see that there are two complexity penalties:

  - ✓ one arising from the BIC approximation to the marginal likelihood,
  - ✓ and the other arising from the prior on $p(\gamma)$.

Bernoulli-Gaussian Model to ℓ_0 Regularization

Bernoulli-Gaussian Model

- Another model used is

$$y_i|\boldsymbol{x}_i, \boldsymbol{w}, \gamma, \sigma^2 \sim \mathcal{N}\left(\sum_j \gamma_j w_j x_{ij}, \sigma^2\right)$$

$$\gamma_j \sim \mathcal{B}er(\pi_0)$$

$$w_j \sim \mathcal{N}(0, \sigma_w^2)$$

- This is called the Bernoulli-Gaussian model. We could also call it the binary mask model, since the γ_j variables are "masking out" w_j.
- Unlike the spike and slab model, we do not integrate out the "irrelevant" coefficients.
- In addition, the binary mask model has the form $\gamma_j \rightarrow \boldsymbol{y} \leftarrow w_j$, whereas the spike and slab model has the form $\gamma_j \rightarrow w_j \rightarrow \boldsymbol{y}$.
- In binary mask model, only $\gamma_j w_j$ can be identified from the likelihood.

- Kuo, L. and B. Mallick (1998). Variable selection for regression models. *Sankhya Series B 60*, 65–81.
- Zhou, H., D. Karakos, S. Khudanpur, A. Andreou, and C. Priebe (2009). On Projections of Gaussian Distributions using Maximum Likelihood Criteria. In *Proc. of the Workshop on Information Theory and its Applications*.
- Soussen, C., J. Iier, D. Brie, and J. Duan (2010). From Bernoulli- Gaussian deconvolution to sparse signal restoration. Technical report, Centre de Recherche en Automatique de Nancy.

Bernoulli-Gaussian Model

- This model can be used to derive an objective function that is widely used in the (non-Bayesian) subset selection literature.

- The joint prior has the form

$$p(\gamma, \boldsymbol{w}) \propto \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \sigma_w^2 \mathbf{I}) \pi_0^{\|\gamma\|_0} (1-\pi_0)^{D-\|\gamma\|_0}$$

- Hence the scaled unnormalized negative log posterior has the form

$$f(\gamma, \boldsymbol{w}) \triangleq -2\sigma^2 \log p(\gamma, \boldsymbol{w}, \boldsymbol{y}|\mathbf{X})$$
$$= \|\boldsymbol{y} - \mathbf{X}(\gamma .* \boldsymbol{w})\|^2 + \frac{\sigma^2}{\sigma_w^2} \|\boldsymbol{w}\|^2 + \lambda \|\gamma\|_0 + \text{const}$$

where

$$\lambda \triangleq 2\sigma^2 \log \frac{1-\pi_0}{\pi_0}$$

ℓ_0 Regularization

- Split $\boldsymbol{w}$ into $\boldsymbol{w}_{-\gamma}$ and $\boldsymbol{w}_{\gamma}$ indexed by the zero & non-zero entries of γ.
- Now consider $\sigma_w^2 \to \infty$, so we do not regularize the non-zero weights (no complexity coming from the marginal likelihood).
- In this case, $f(\gamma, \boldsymbol{w}) = \|\boldsymbol{y} - \mathbf{X}(\gamma .* \boldsymbol{w})\|^2 + \frac{\sigma^2}{\sigma_w^2}\|\boldsymbol{w}\|^2 + \lambda\|\gamma\|_0 + \text{const}$ becomes

$$f(\gamma, \boldsymbol{w}) = \left\|\boldsymbol{y} - \mathbf{X}_{\gamma}\boldsymbol{w}_{\gamma}\right\|_2^2 + \lambda\|\boldsymbol{\gamma}\|_0$$

- This is similar to the BIC objective.
- Instead of keeping track of the bit vector γ, we can define the set of relevant variables to be the support, or set of non-zero entries, of $\boldsymbol{w}$.
- Then we can rewrite the above equation as

$$f(\boldsymbol{w}) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_0$$

- This is called ℓ_0 regularization. The discrete optimization problem over $\boldsymbol{\gamma} \in \{0,1\}^D$ is now transformed to a continuous one over $\boldsymbol{w} \in \mathbb{R}^D$.

Algorithms for Variable Selection

Evaluating the Marginal Likelihood

- Since there are 2^D models, we cannot explore the full posterior, or find the globally optimal model.
- Instead we will have to resort to heuristics of one form or another.
- All of the methods we will discuss involve searching through the space of models, and evaluating the cost $f(\gamma)$ at each point.
- This requires fitting the model, or evaluating its marginal likelihood.
- This is sometimes called the wrapper method - we "wrap" our search for the best model around a generic model-fitting procedure.
- In order to make wrapper methods efficient, it is important that we can quickly evaluate the score function for some new model, γ', given the score of a previous model, γ.

Evaluating the Marginal Likelihood

- This can be done provided we can efficiently update the sufficient statistics needed to compute $f(\gamma)$.
- This is possible provided γ' only differs from γ in one bit (corresponding to adding or removing a single variable), and provided $f(\gamma)$ only depends on the data via $\mathbf{X}_\gamma$.
- In this case, we can use rank-one matrix updates / downdates to efficiently compute $\mathbf{X}_{\gamma'}^T\mathbf{X}_{\gamma'}$ from $\mathbf{X}_\gamma^T\mathbf{X}_\gamma$.
- These updates are usually applied to the QR decomposition of $\boldsymbol{X}$.
- If we use the ℓ_0 regularized objective, we can exploit properties of least squares to derive various efficient greedy forwards search methods, some of which we summarize below.

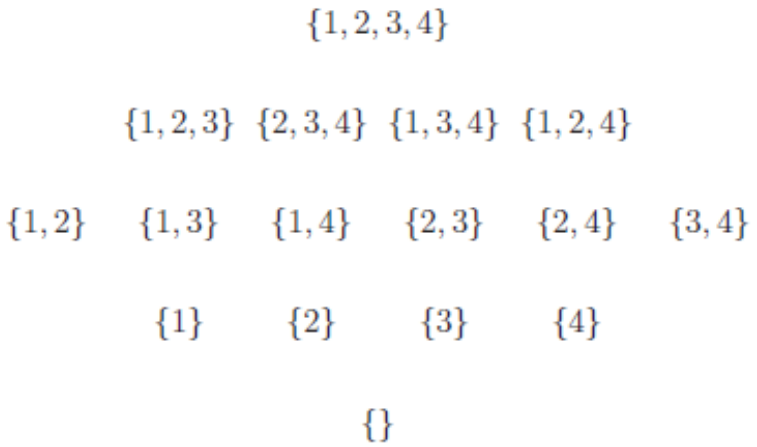
- Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer.
- Miller, A. (2002). *Subset selection in regression*. Chapman and Hall. 2nd edition.
- Schniter, P., L. C. Potter, and J. Ziniel (2008). Fast Bayesian Matching Pursuit: Model Uncertainty and Parameter Estimation for Sparse Linear Models. Technical report, U. Ohio. Submitted to IEEE Trans. On Signal Processing.

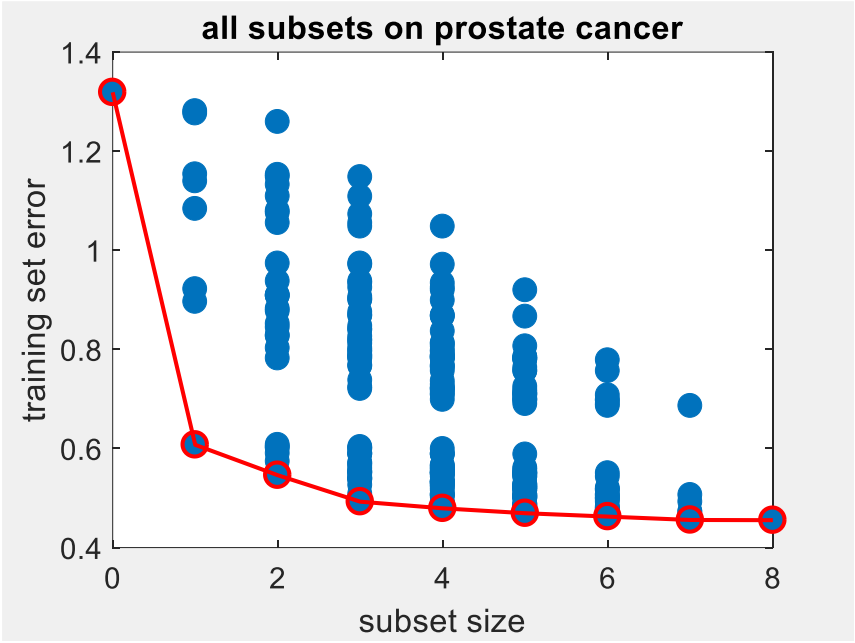
Single Best Replacement (SBR)

- The simplest method is to use greedy hill climbing, where at each step, we define the neighborhood of the current model to be all models than can be reached by flipping a single bit of γ, i.e., for each variable, if it is currently out of the model, we consider adding it, and if it is currently in the model, we consider removing it.

- Soussen, C., J. Iier, D. Brie, and J. Duan (2010). From Bernoulli- Gaussian deconvolution to sparse signal restoration. Technical report, Centre de Recherche en Automatique de Nancy.

Single Best Replacement (SBR)





prostateSubsets.m
from PMTK3

- (a) A lattice of subsets of {1, 2, 3, 4}.
- (b) Residual sum of squares versus subset size, on the prostate cancer data set. The lower envelope is the best RSS achievable for any set of a given size.

We are essentially moving through the lattice of subsets, adding or removing until no improvement is possible.

- Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer.

Orthogonal Least Squares

- If we set $\lambda = 0$ in $f(\boldsymbol{w}) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_0$, so there is no complexity penalty, there will be no reason to perform deletion steps. In this case, the SBR algorithm is equivalent to **orthogonal least squares**, which in turn is equivalent to **greedy forwards selection**.

- In this algorithm, we start with the empty set and add the best feature at each step. The error will go down monotonically with $\|\boldsymbol{\gamma}\|_0$.

- We can pick the next best feature j^* to add to the current set γ_t by solving

$$j^* = \underset{j \notin \gamma_t}{\operatorname{argmin}} \min_{\boldsymbol{w}} ||\boldsymbol{y} - (\boldsymbol{X}_{\gamma_t \cup j})\boldsymbol{w}||_2$$

- We then update the active set by setting $\gamma_{t+1} = \gamma_t \cup \{j\}$. To choose the next feature to add at step t, we need to solve $D - D_t$ least squares problems at step t, where $D_t = |\gamma_t|$ is the cardinality of the current active set.

- Having chosen the best feature to add, we need to solve an additional least squares problem to compute $\boldsymbol{w}_{t+1}$.

  - Chen, S. and J. Wigger (1995, July). Fast orthogonal least squares algorithm for efficient subset model selection. *IEEE Trans. Signal Processing 3*(7), 1713–1715.

Orthogonal Matching Pursuits

- Orthogonal least squares is somewhat expensive.
- A simplification is to "freeze" the current weights at their current value, and then to pick the next feature to add by solving

$$j^* = \arg\min_{j \notin \gamma_t} \min_{\beta} \left\| \boldsymbol{y} - \mathbf{X}\boldsymbol{w}_t - \beta \mathbf{x}_{:,j} \right\|^2$$

- This inner optimization is easy to solve; we simply set $\beta = \frac{\mathbf{x}_{:,j}^T \boldsymbol{r}_t}{\left\| \mathbf{x}_{:,j} \right\|^2}$, where $r_t = \boldsymbol{y} - \mathbf{X}\boldsymbol{w}_t$ is the current residual vector.
- If the columns are unit norm, we have

$$j^* = \arg\max \mathbf{x}_{:,j}^T \boldsymbol{r}_t$$

- We are just looking for the column that is most correlated with the current residual.

- Mallat, S., G. Davis, and Z. Zhang (1994, July). Adaptive time frequency decompositions. *SPIE Journal of Optical Engineering 33*, 2183–2919.
- Blumensath, T. and M. Davies (2007). On the difference between Orthogonal Matching Pursuit and Orthogonal Least Squares. Technical report, U. Edinburgh.

Orthogonal Matching Pursuits

- We then update the active set, and compute the new least squares estimate $\boldsymbol{w}_{t+1}$ using $\mathbf{X}_{\gamma_t+1}$.
- This method is called orthogonal matching pursuits or OMP.
- This only requires one least squares calculation per iteration and so is faster than orthogonal least squares, but is not quite as accurate.
- An even more aggressive approximation is to just greedily add the feature that is most correlated with the current residual. This is called matching pursuits.
- This is also equivalent to a method known as least squares boosting.

- Mallat, S., G. Davis, and Z. Zhang (1994, July). Adaptive time frequency decompositions. *SPIE Journal of Optical Engineering 33*, 2183–2919.
- Blumensath, T. and M. Davies (2007). On the difference between Orthogonal Matching Pursuit and Orthogonal Least Squares. Technical report, U. Edinburgh.
- Mallat, S. and Z. Zhang (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing 41*(12), 3397–3415.

Backwards Selection

- This starts with all variables in the model (the so-called saturated model), and then deletes the worst one at each step.
- This is equivalent to performing a greedy search from the top of the lattice downwards.
- This can give better results than a bottom-up search, since the decision about whether to keep a variable or not is made in the context of all the other variables that might depend on it.
- However, this method is typically infeasible for large problems, since the saturated model will be too expensive to fit.

- Mallat, S. and Z. Zhang (1993). Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing 41*(12), 3397–3415.

Forwards-Backwards (FoBa)

- This method is similar to the single best replacement algorithm presented earlier, except it uses an orthogonal matching pursuits OMP-like approximation when choosing the next move to make.

- Zhang, T. (2008). Adaptive Forward- Backward Greedy Algorithm for Sparse Learning with Linear Models. In *NIPS*.
- Moghaddam, B., A. Gruber, Y. Weiss, and S. Avidan (2008). Sparse regression as a sparse eigenvalue problem. In *Information Theory & Applications Workshop (ITA'08)*.

Bayesian Matching Pursuits

- This algorithm is similar to (orthogonal matching pursuits) OMP except it uses a Bayesian marginal likelihood scoring criterion (under a spike and slab model) instead of a least squares objective.

- In addition, it uses a form of beam search to explore multiple paths through the lattice at once.

- Schniter, P., L. C. Potter, and J. Ziniel (2008). Fast Bayesian Matching Pursuit: Model Uncertainty and Parameter Estimation for Sparse Linear Models. Technical report, U. Ohio. Submitted to IEEE Trans. On Signal Processing.

Stochastic Search

- If we want to approximate the posterior, rather than just computing a mode (e.g. because we want to compute marginal inclusion probabilities), one option is to use MCMC.
- The standard approach is to use Metropolis Hastings, where the proposal distribution just flips single bits.
- This enables us to efficiently compute $p(\gamma'|\mathcal{D})$ given $p(\gamma|\mathcal{D})$.
- The probability of a state (bit configuration) is estimated by counting how many times the random walk visits this state.
- However, in a discrete state space, MCMC is needlessly inefficient, since we can compute the (unnormalized) probability of a state directly using $p(\gamma, \mathcal{D}) = \exp\big(-f(\gamma)\big)$.
  - Thus, there is no need to ever revisit a state.

Stochastic Search

- A much more efficient alternative is to use some kind of stochastic search algorithm, to generate a set $\mathcal{S}$ of high scoring models.

- Then, we make the following approximation

$$p(\gamma|\mathcal{D}) \approx \frac{\exp\left(-f(\gamma)\right)}{\sum_{\gamma' \in \mathcal{S}} \exp\left(-f(\gamma')\right)}$$

- O'Hara, R. and M. Sillanpaa (2009). A Review of Bayesian Variable Selection Methods: What, How and Which. *Bayesian Analysis 4*(1), 85– 118.
- Bottolo, L. and S. Richardson (2010). Evolutionary stochastic search. *Bayesian Analysis 5*(3), 583–618.
- Heaton, M. and J. Scott (2009). Bayesian computation and the linear model. Technical report, Duke.

EM and Variational Inference

- It is tempting to apply EM to the spike and slab model, which has the form $\gamma_j \rightarrow w_j \rightarrow \boldsymbol{y}$.
  - We can compute $p(\gamma_j = 1|w_j)$ in the E-step
  - We optimize $\boldsymbol{w}$ in the M-step.
- This will not work, because when we compute $p(\gamma_j = 1|w_j)$, we are comparing a delta-function, $\delta_0(w_j)$, with $\mathcal{N}(w_j|0, \sigma^2\sigma_w^2)$.
  - We can replace $\delta_0(w_j)$ with a narrow Gaussian, and then the E step amounts to classifying w_j under the two possible Gaussian models.
  - However, this is likely to suffer from severe local minima.
- Can also apply EM to the Bernoulli-Gaussian model $\gamma_j \rightarrow \boldsymbol{y} \leftarrow w_j$. However, $p(\boldsymbol{\gamma}|\mathcal{D}, \boldsymbol{w})$ is intractable as all bits become correlated.
  - It is possible to derive a mean field approximation $\prod_j q(\gamma_j)q(w_j)$.

- Huang, J., Q. Morris, and B. Frey (2007). Bayesian inference of MicroRNA targets from sequence and expression data. *J. Comp. Bio.*.
- Rattray, M., O. Stegle, K. Sharp, and J. Winn (2009). Inference algorithms and learning theory for Bayesian sparse factor analysis. In *Proc. Intl. Workshop on Statistical- Mechanical Informatics*.

ℓ_1 Regularization

ℓ_1 Regularization

- When we have many variables, it is computationally difficult to find the posterior mode of $p(\gamma|\mathcal{D})$.
- Although greedy algorithms often work well, they can get stuck in local optima.
- Part of the problem is due to the fact that the γ_j variables are discrete.
- In the optimization community, it is common to relax hard constraints of this form by replacing discrete variables with continuous variables.
- We can do this by replacing the spike-and-slab style prior that assigns finite probability mass to the event that $w_j = 0$, to continuous priors that **encourage** $w_j = 0$ by putting a lot of probability density near the origin, such as a zero-mean Laplace distribution.

- Zhang, T. (2008). Adaptive Forward- Backward Greedy Algorithm for Sparse Learning with Linear Models. In *NIPS*.

ℓ_1 Regularization

- Consider a prior of the form

$$p(\boldsymbol{w}|\lambda) = \prod_{j=1}^{D} \mathcal{L}ap\left(w_j|0,1/\lambda\right) \propto \prod_{j=1}^{D} \exp\left(-\lambda|w_j|\right)$$

- We will use a uniform prior on the offset term, $p(w_0) \propto 1$.
- The penalized negative log likelihood has the form

$$f(\boldsymbol{w}) = -\log p(\mathcal{D}|\boldsymbol{w}) - \log p(\boldsymbol{w}|\lambda) = \text{NLL}(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_1$$

- $\|\boldsymbol{w}\|_1$ is the ℓ_1 norm and defined as

$$\|\boldsymbol{w}\|_1 = \sum_{j=1}^{D} |w_j|$$

- For suitably large λ, the estimate $\hat{\boldsymbol{w}}$ will be sparse.
- Indeed, this can be thought of as a convex approximation to the non-convex ℓ_0 objective $\arg\min_{\mathbf{w}} \text{NLL}(\boldsymbol{w}) + \lambda\|\boldsymbol{w}\|_0$.

ℓ_1 Regularization

- In the case of linear regression, ℓ_1 objective becomes

$$\sum_{i=1}^{N} \frac{1}{2\sigma^2}\left(y_i - (w_0 + \boldsymbol{w}^T \boldsymbol{x}_i)\right)^2 + \lambda \|\boldsymbol{w}\|_1$$

or

$$f(\boldsymbol{w}) = \mathrm{RSS}(\boldsymbol{w}) + \lambda' \|\boldsymbol{w}\|_1$$

where $\lambda' = 2\lambda\sigma^2$.

- This method is known as basis pursuit denoising or BPDN.
- In general, the technique of putting a zero-mean Laplace prior on the parameters and performing MAP estimation is called ℓ_1 regularization.
- It can be combined with any convex or non-convex NLL term.
- Many different algorithms have been devised for such problems.

- Chen, S. and J. Goodman (1998). An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Dept. Comp. Sci., Harvard.

Lasso Regression

- We have the following non-smooth objective function

$$\min_{\boldsymbol{w}} \mathrm{RSS}(\boldsymbol{w}) + \lambda' \|\boldsymbol{w}\|_1$$

- We can rewrite this (dual form) as a constrained but smooth objective (a quadratic function with linear constraints):

$$\min_{\boldsymbol{w}} \mathrm{RSS}(\boldsymbol{w}) \qquad \text{s.t.} \|\boldsymbol{w}\|_1 \leq B$$

where B is the upper bound on the ℓ_1-norm of the weights.

- A small (tight) bound corresponds to a large penalty and vice versa.

- This is known as LASSO, which stands for "least absolute shrinkage and selection operator".

- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B 58*(1), 267 288.

Lasso Regression

- Similarly, we can write ridge regression

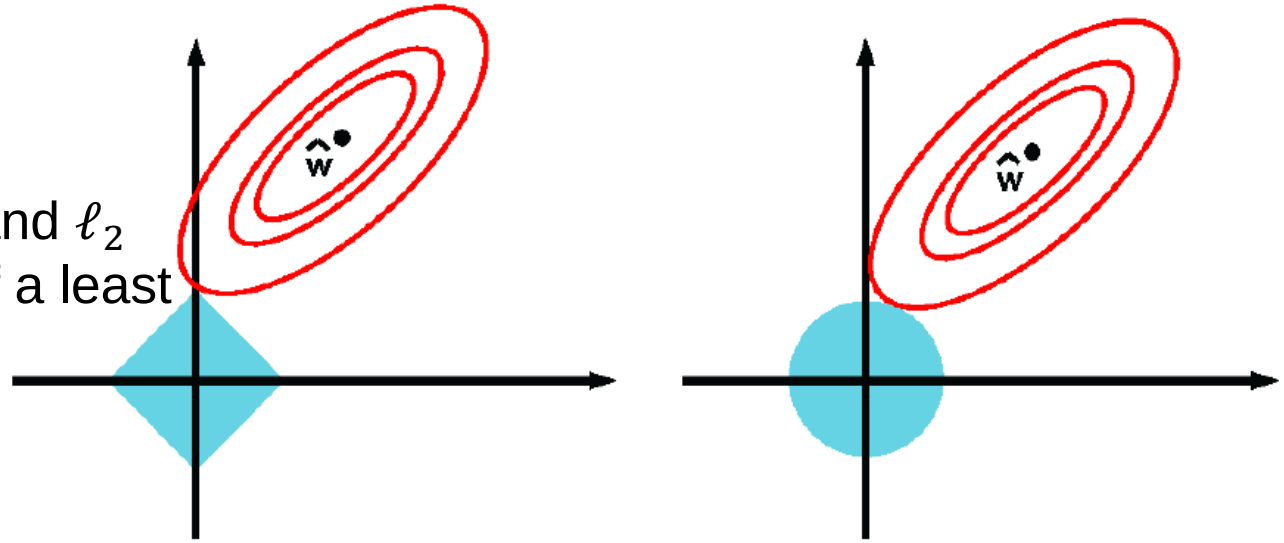
$$\min_{\boldsymbol{w}} \mathrm{RSS}(\boldsymbol{w}) + \lambda'' \|\boldsymbol{w}\|_2^2$$

- This can be written as

$$\min_{\boldsymbol{w}} \mathrm{RSS}(\boldsymbol{w}) \quad s.t. \|\boldsymbol{w}\|_2^2 \leq B$$

- We plot the contour of the RSS objective function, as well as the contours of ℓ_1 and ℓ_2 surfaces

Illustration of ℓ_1 (left) and ℓ_2 (right) regularization of a least squares problem.



Lasso Regression

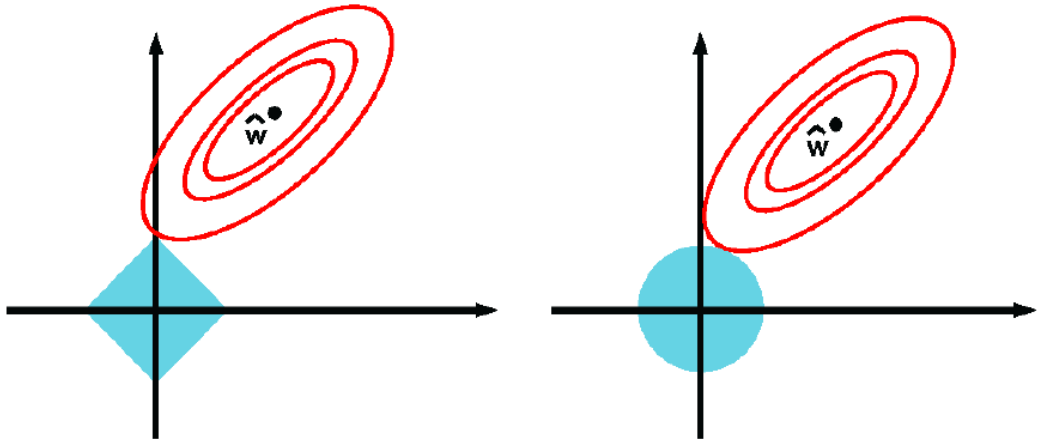
There are three major differences between ridge and lasso:

- The sharp, non-differentiable corners of the ℓ_1 –ball produce parsimonious models for sufficiently large values of λ.
- The lack of rotational invariance limits the use of the singular value theory (SVD) that is so useful in unpenalized and ridge regression.
- Lasso lacks an analytic solution making both computation and theoretical results more difficult.

ℓ_1 Regularization Yields Sparse Solutions

- From the theory of constrained optimization, we know that the optimal solution occurs at the point where the lowest level set of the objective function intersects the constraint surface.

- It should be geometrically clear that as we relax the constraint B, we "grow" the ℓ_1 ball until it meets the objective.
  - The corners of the ball are more likely to intersect the ellipse than one of the sides, especially in high dimensions, because the corners "stick out" more
  - The corners correspond to sparse solutions, which lie on the coordinate axes



- By contrast, when we grow the ℓ_2 ball, it can intersect the objective at any point; there are no "corners", so there is no preference for sparsity.

ℓ_1 Regularization Yields Sparse Solutions

- To see this another away, notice that, with ridge regression, the prior cost of a sparse solution such as $\boldsymbol{w} = (1,0)$ is the same as the cost of a dense solution, such as $\boldsymbol{w} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$

$$\|(1,0)\|_2 = \left\|\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\right\|_2 = 1$$

- However, for lasso, setting $\boldsymbol{w} = (1,0)$ is cheaper than setting $\boldsymbol{w} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$.

$$\|(1,0)\|_1 = 1 < \left\|\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\right\|_2 = \sqrt{2}$$

- The most rigorous way to see that ℓ_1 regularization results in sparse solutions is to examine conditions that hold at the optimum.

Lasso Models Vs Other Models

- Ordinary least squares and ridge regression have *analytic solutions*, we can write down an explicit formula for what the estimators are.

- Generalized linear models have only *numerical solutions* with iterative methods. We have algorithms (IRLS) that, when run sufficiently long should yield a solution with a reasonable accuracy.

- The lasso falls somewhere in-between these two cases, as it has a *direct numerical solution*.
  - We cannot write out an explicit analytic form, but
  - the algorithm is not iterative and would yield the exact solution given a machine with infinite precision.

Optimality Conditions for Lasso

- The lasso objective has the form

$$f(\theta) = \text{RSS}(\theta) + \lambda \|\boldsymbol{w}\|_1$$

- Unfortunately, the $\|\boldsymbol{w}\|_1$ term is not differentiable whenever $w_j = 0$.

- This is an example of a non-smooth optimization problem.

- To handle non-smooth functions, we need to extend the notion of a derivative.

- We define a subderivative or subgradient of a (convex) function $f: \mathcal{I} \to \mathbb{R}$ at a point θ_0 to be a scalar g such that

$$f(\theta) - f(\theta_0) \geq g(\theta - \theta_0) \;\; \forall \, \theta \in \mathcal{I}$$

  where $\mathcal{I}$ is some interval containing θ_0.

The SubDifferential

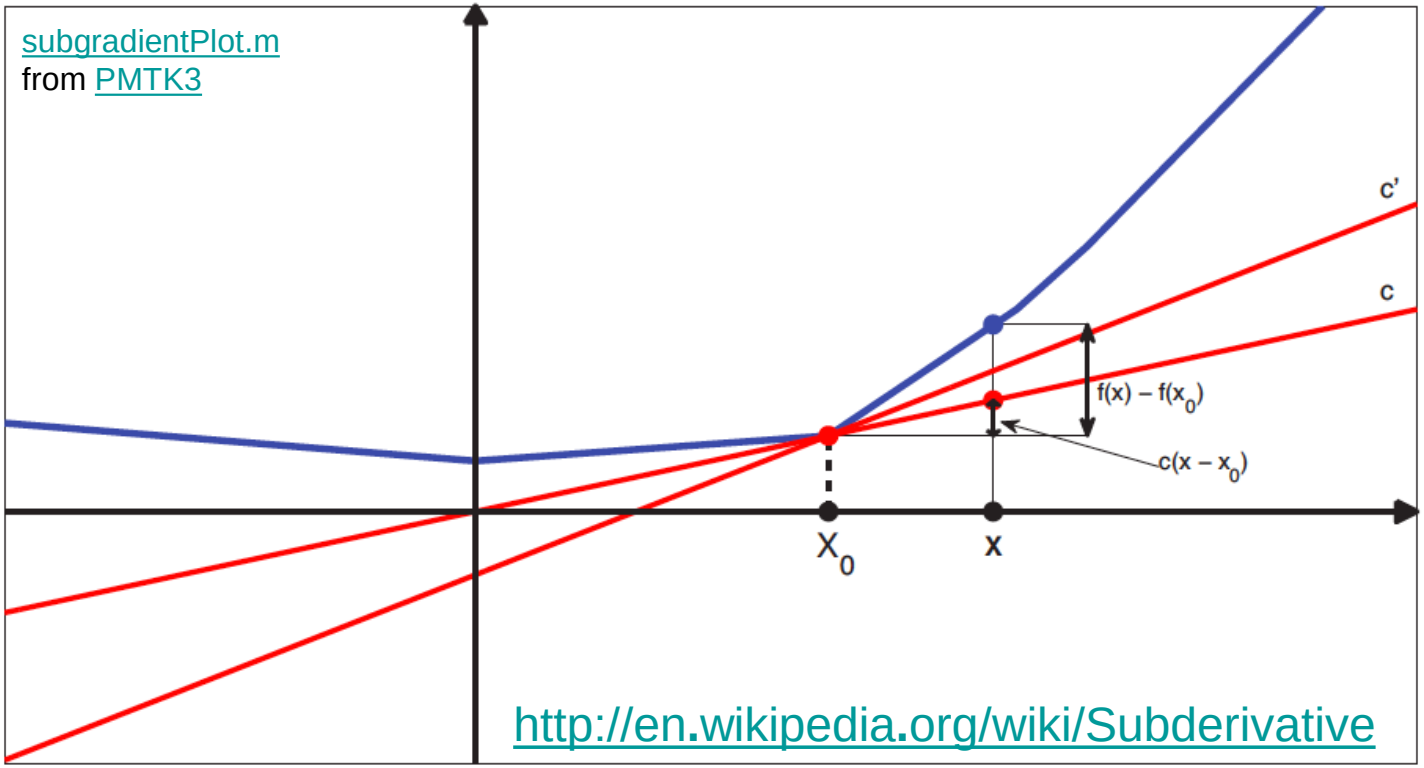


Illustration of some sub-derivatives of a function at point x_0.

Subderivatives

- We define the set of subderivatives as the interval $[a, b]$ where a and b are the one-sided limits

$$a = \lim_{\theta \to \theta_0^-} \frac{f(\theta) - f(\theta_0)}{\theta - \theta_0}, b = \lim_{\theta \to \theta_0^+} \frac{f(\theta) - f(\theta_0)}{\theta - \theta_0}$$

- The set $[a, b]$ of all subderivatives is called the subdifferential of the function f at θ_0 and is denoted as $\partial f(\theta)|_{\theta_0}$.

- For example, in the case of the absolute value function $f(\theta) = |\theta|$, the subderivative is

$$\partial f(\theta)\Big|_{\theta_0} = \begin{cases} \{-1\} & \text{if } \theta < 0 \\ [-1,1] & \text{if } \theta = 0 \\ \{+1\} & \text{if } \theta > 0 \end{cases}$$

- If the function is everywhere differentiable, then $\partial f(\theta) = \left\{\frac{df(\theta)}{d\theta}\right\}$.

Subdifferential of a Convex Function

We will use three properties of the subdifferential of a convex function f:

1. The gradient $\nabla(f)$ exists at the point x_0 if and only if $\partial(f)(x_0)$ is equal to a single value, which is equal to $\nabla(f)(x_0)$

2. For every point x_0, the set $\partial(f)(x_0)$ is a nonempty closed interval $[a;\ b]$

3. The point x_0 is a global minimum of f if and only if the subdifferential contains zero; in other words, $0 \in \partial(f)(x_0)$

Optimality Conditions for Lasso

- We sketch out how the algorithm works in the simple case where $\boldsymbol{X}^T\boldsymbol{X}$ is the identity matrix.
- Let us calculate the subdifferential and determine what solution $\boldsymbol{w}$ give us $\partial(f)(\boldsymbol{w})$ that contain 0.
- To simplify matters, assume that $\boldsymbol{X}^T\boldsymbol{X}$ is equal to the identity matrix. We will also add a factor of 2 to the penalty, i.e. $2\lambda\|\boldsymbol{w}\|_1$.
- The objective in this case is given by:

$$f(\boldsymbol{w}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + 2\lambda\|\boldsymbol{w}\|_1 = \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{w}^T\boldsymbol{w} - 2\boldsymbol{y}^T\boldsymbol{X}\boldsymbol{w} + 2\lambda\|\boldsymbol{w}\|_1$$

- We know that the gradient of the front terms is just $2\boldsymbol{w} - 2\boldsymbol{X}^T\boldsymbol{y}$, and the subdifferential of the ℓ_1 term w.r.t. w_j is $\begin{cases} \{-2\lambda\} & \text{if } w_j < 0 \\ [-2\lambda, 2\lambda] & \text{if } w_j = 0 \\ \{+2\lambda\} & \text{if } w_j > 0 \end{cases}$

- Efron, Bradley, I. Johnstone, T. Hastie, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*(2),407–499.

Optimality Conditions for Lasso

- Thus the j-th component of the subdifferential is:

$$\partial(f)(w_j)=\begin{cases}\left\{2w_j-2\boldsymbol{x}_j^T\boldsymbol{y}+2\lambda\right\} & \text{if } w_j>0\\ [-2\lambda,+2\lambda]-2\boldsymbol{x}_j^T\boldsymbol{y} & \text{if } w_j=0\\ \left\{2w_j-2\boldsymbol{x}_j^T\boldsymbol{y}-2\lambda\right\} & \text{if } w_j<0\end{cases}$$

- Need to find what will make all of these zero for all j's.

- Efron, Bradley, I. Johnstone, T. Hastie, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*(2),407–499.

Optimality Conditions for Lasso

$$\partial(f)(w_j)=\begin{cases}\{2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} + 2\lambda\} & \text{if } w_j > 0\\ [-2\lambda, +2\lambda] - 2\boldsymbol{x}_j^T\boldsymbol{y} & \text{if } w_j = 0\\ \{2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} - 2\lambda\} & \text{if } w_j < 0\end{cases}$$

- If $w_j > 0$, we need the following to hold:

$$2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} + 2\lambda = 0 \rightarrow w_j = \boldsymbol{x}_j^T\boldsymbol{y} - \lambda$$

- So w_j is a linear function of λ, with an intercept of $\boldsymbol{x}_j^T\boldsymbol{y}$ and a slope of -1. However this only holds when:

$$\boldsymbol{x}_j^T\boldsymbol{y} - \lambda > 0 \rightarrow \boldsymbol{x}_j^T\boldsymbol{y} > \lambda$$

- So, for all $0 < \lambda < \boldsymbol{x}_j^T\boldsymbol{y}$. So clearly w_j can only be positive if $\boldsymbol{x}_j^T\boldsymbol{y} > 0$.

- Efron, Bradley, I. Johnstone, T. Hastie, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*(2),407–499.

Optimality Conditions for Lasso

$$\partial(f)(w_j)=\begin{cases}\{2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} + 2\lambda\} & \text{if } w_j > 0\\ [-2\lambda, +2\lambda] - 2\boldsymbol{x}_j^T\boldsymbol{y} & \text{if } w_j = 0\\ \{2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} - 2\lambda\} & \text{if } w_j < 0\end{cases}$$

- If $w_j < 0$, we need the following to hold:

$$2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} - 2\lambda = 0 \rightarrow w_j = \boldsymbol{x}_j^T\boldsymbol{y} + \lambda$$

- So w_j is a linear function of λ, with an intercept of $\boldsymbol{x}_j^T\boldsymbol{y}$ and a slope of $+1$. However this only holds when:

$$\boldsymbol{x}_j^T\boldsymbol{y} + \lambda < 0 \rightarrow \boldsymbol{x}_j^T\boldsymbol{y} < -\lambda$$

- So, for all $\lambda < -\boldsymbol{x}_j^T\boldsymbol{y}$. So clearly w_j can only be negative if $\boldsymbol{x}_j^T\boldsymbol{y} < 0$.

- Combine the two previous conditions as: $w_j = \boldsymbol{x}_j^T\boldsymbol{y} - sign(\boldsymbol{x}_j^T\boldsymbol{y})\lambda$.

Optimality Conditions for Lasso

$$\partial(f)(w_j)=\begin{cases}\{2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} + 2\lambda\} & \text{if } w_j > 0\\ [-2\lambda, +2\lambda] - 2\boldsymbol{x}_j^T\boldsymbol{y} & \text{if } w_j = 0\\ \{2w_j - 2\boldsymbol{x}_j^T\boldsymbol{y} - 2\lambda\} & \text{if } w_j < 0\end{cases}$$

- If $w_j = 0$, we need the following to hold:

$$0 \in [-2\lambda, +2\lambda] - 2\boldsymbol{x}_j^T\boldsymbol{y} = 0 \rightarrow \begin{cases}-2\lambda - 2\boldsymbol{x}_j^T\boldsymbol{y} < 0\\ 0 < 2\lambda - 2\boldsymbol{x}_j^T\boldsymbol{y}\end{cases} \rightarrow \begin{cases}-\boldsymbol{x}_j^T\boldsymbol{y} < \lambda\\ \lambda > \boldsymbol{x}_j^T\boldsymbol{y}\end{cases} \rightarrow \lambda > \left|\boldsymbol{x}_j^T\boldsymbol{y}\right|$$

- Finally the overall solution is:

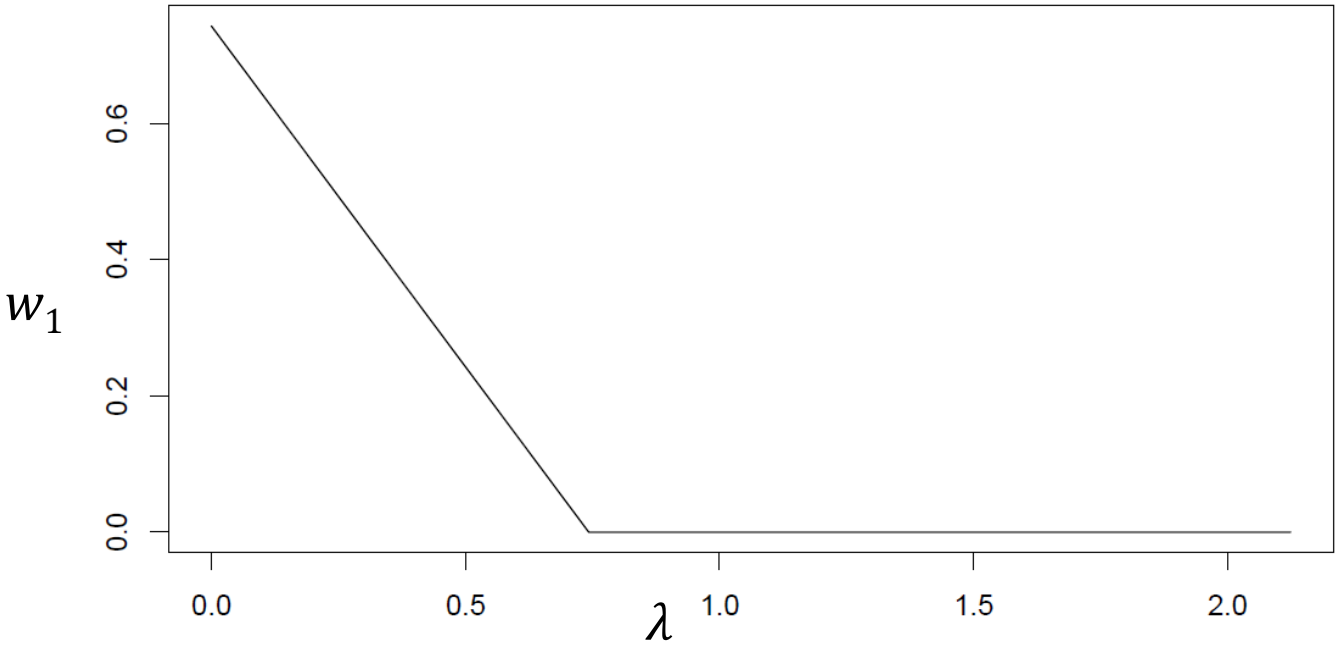
$$w_j = \begin{cases}0 & if \quad \lambda > \left|\boldsymbol{x}_j^T\boldsymbol{y}\right|\\ \boldsymbol{x}_j^T\boldsymbol{y} - sign\left(\boldsymbol{x}_j^T\boldsymbol{y}\right)\lambda & if \quad \lambda \le \left|\boldsymbol{x}_j^T\boldsymbol{y}\right|\end{cases}$$

Example

- Construct a data set with $\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{I}$, use an orthogonal polynomial basis of order 5, and for data generation use $\boldsymbol{w}^T = \{(1,0,1,0,0)\}$.

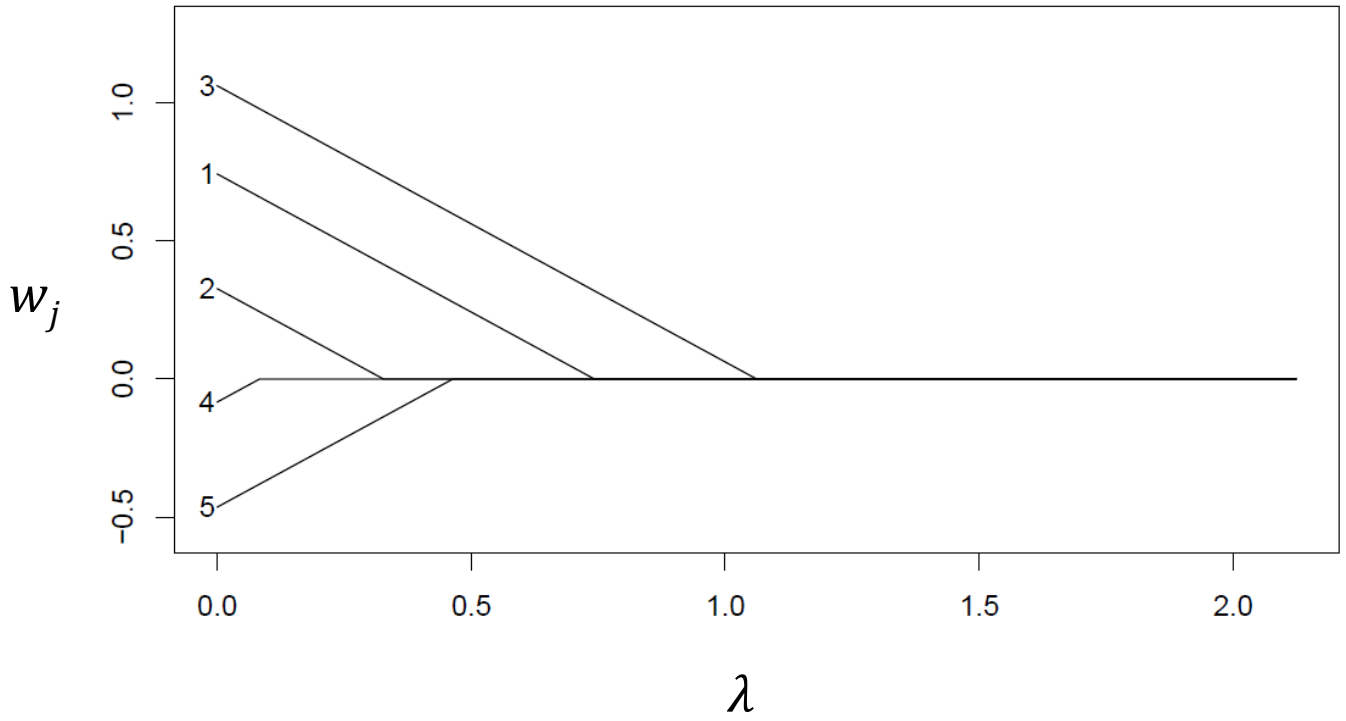
- Constructing a sequence of λ values to which we want to fit the model. We know the non-zero solutions lie between 0 and $\|\boldsymbol{X}^T\boldsymbol{y}\|_\infty$ (we pick twice that range)

- We plot w_1 as a function of λ.



Example

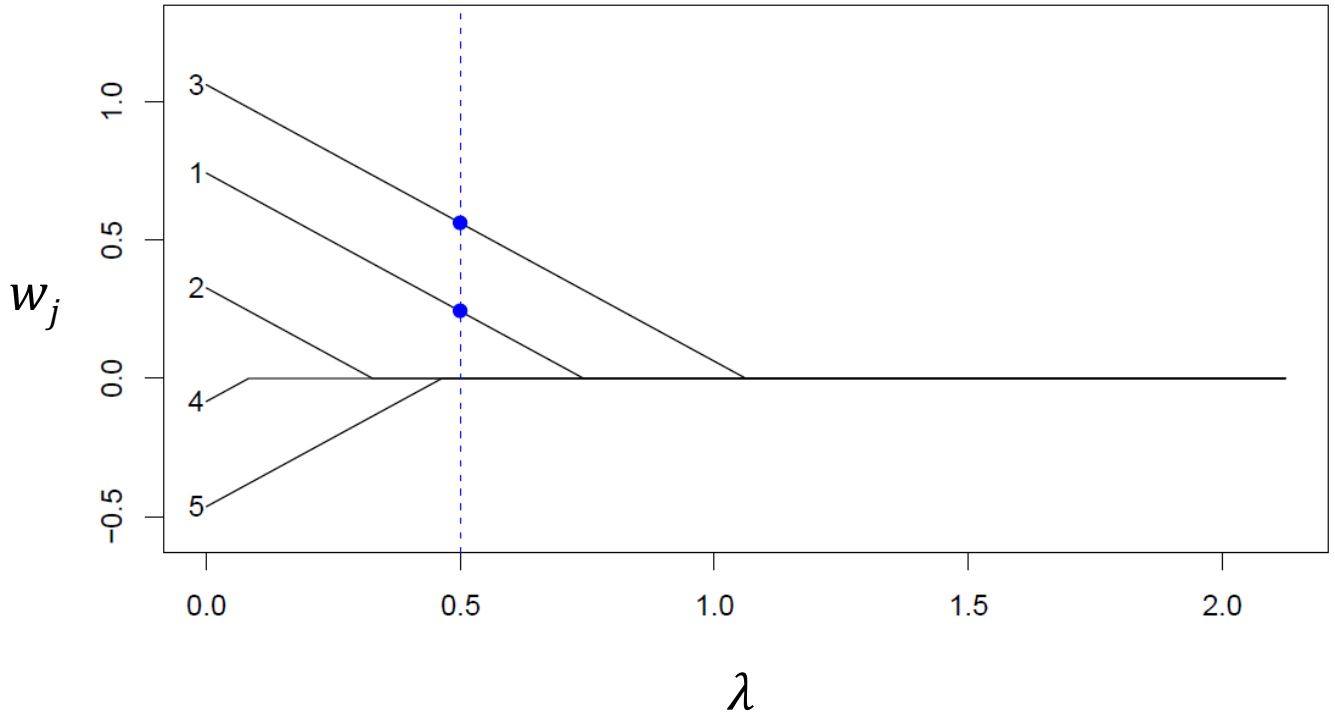
- The full set of the w_j as a function of λ are shown below:



- Recall that only elements 1 and 3 were actually non-zero. Note what happens when you plot the solution for $\lambda = 0.5$.

Example

- Solution for $\lambda = 0.5$



Optimality Conditions for Lasso

- By analogy to the standard calculus result, one can show that the point $\hat{\theta}$ is a local minimum if $0 \in \partial f(\theta)|_{\hat{\theta}}$.

- Applying this concept to lasso for the more general case,

$$\frac{\partial}{\partial w_j}\text{RSS}(\boldsymbol{w}) = a_j w_j - c_j$$

$$a_j = 2\sum_{i=1}^{n} x_{ij}^2$$

$$c_j = 2\sum_{i=1}^{n} x_{ij}\left(y_i - \boldsymbol{w}_{-j}^T \boldsymbol{x}_{i,-j}\right)$$

where $\boldsymbol{w}_{-j}$ is $\boldsymbol{w}$ without component j (similar for $\boldsymbol{x}_{i,-j}$)

- We see that c_j is (proportional to) the correlation between the j'th feature $\boldsymbol{x}_{:,j}$ and the residual due to the other features.

Optimality Conditions for Lasso

- Hence the magnitude of c_j is an indication of how relevant feature j is for predicting $\boldsymbol{y}$.

- Adding in the penalty term, we find that the subderivative is given by

$$\partial_{w_j} f(\boldsymbol{w}) = \left(a_j w_j - c_j\right) + \lambda \partial_{w_j} \|\boldsymbol{w}\|_1$$

$$\partial_{w_j} f(\boldsymbol{w}) = \begin{cases} \{a_j w_j - c_j - \lambda\} & \text{if } w_j < 0 \\ [-c_j - \lambda, -c_j + \lambda] & \text{if } w_j = 0 \\ \{a_j w_j - c_j + \lambda\} & \text{if } w_j > 0 \end{cases}$$

- We can write this in a more compact fashion as follows

$$\mathbf{X}^T(\mathbf{X}\boldsymbol{w} - \boldsymbol{y})_j \in \begin{cases} \{-\lambda\} & \text{if } w_j < 0 \\ [-\lambda, \lambda] & \text{if } w_j = 0 \\ \{\lambda\} & \text{if } w_j > 0 \end{cases}$$

Optimality Conditions for Lasso

- Depending on the value of c_j, the solution to $\partial_{w_j} f(\boldsymbol{w}) = 0$ can occur at three different values of w_j.
  - If $c_j < -\lambda$, so the feature is strongly negatively correlated with the residual, then the subgradient is zero at $\hat{w}_j = \frac{c_j + \lambda}{a_j} < 0$
  - If $c_j \in [-\lambda, \lambda]$, so the feature is only weakly correlated with the residual, then the subgradient is zero at $\hat{w}_j = 0$.
  - If $c_j > \lambda$, so the feature is strongly positively correlated with the residual, then the subgradient is zero at $\hat{w}_j = \frac{c_j - \lambda}{a_j} > 0$
- In summary, we have

$$\hat{w}_j(c_j) = \begin{cases} \dfrac{(c_j + \lambda)}{a_j} & if \;\; c_j < -\lambda \\ 0 & if \; c_j \in [-\lambda, +\lambda] \\ (c_j - \lambda)/a_j & if \; c_j > \lambda \end{cases}$$

Optimality Conditions for Lasso

- We can write this as follows

$$\widehat{w}_j = \text{soft}\left(\frac{c_j}{a_j}; \frac{\lambda}{a_j}\right)$$

$$a_j = 2\sum_{i=1}^{n} x_{ij}^2$$

$$c_j = 2\sum_{i=1}^{n} x_{ij}\left(y_i - \boldsymbol{w}_{-j}^T \boldsymbol{x}_{i,-j}\right)$$

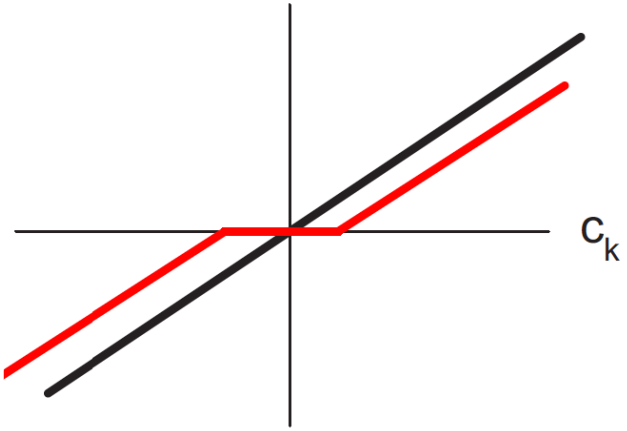
where

$$\text{soft}(a;\delta) \triangleq sign(a)(|a| - \delta)_+$$

and

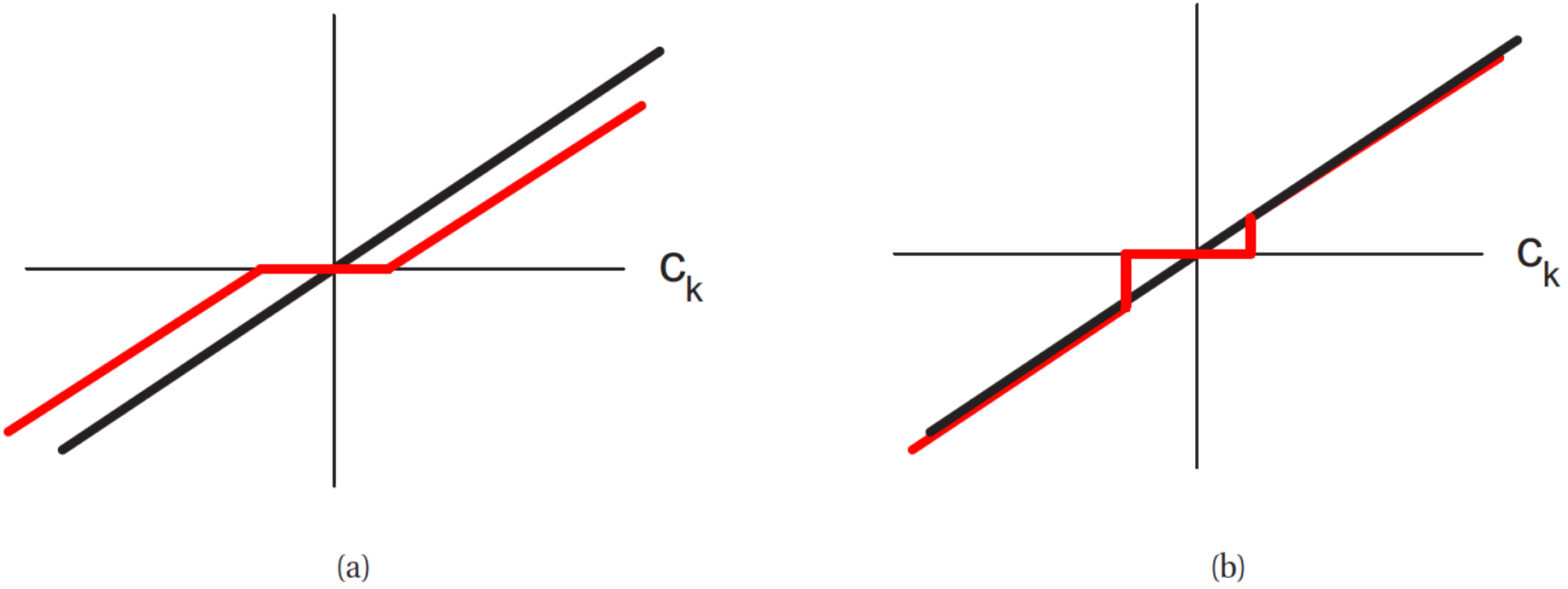
$$x_+ = \max(x, 0)$$

is the positive part of x.



- This is called soft thresholding.
- We illustrate it with a figure.
  - The black line is the line $w_j = c_j/a_j$ corresponding to the least squares fit.
  - The red line, which represents the regularized estimate $\widehat{w}_j(c_j)$, shifts the black line down (or up) by λ, except when $-\lambda \leq c_j \leq \lambda$, in which case it sets $w_j = 0$.

Optimality Conditions for Lasso



- Left: soft thresholding. The flat region is the interval $[-\lambda, +\lambda]$.
- Right: hard thresholding.

- ✓ The black line is the line $w_j = c_j / a_j$ corresponding to the least squares fit.
- ✓ The red line, which represents the regularized estimate $\widehat{w}_j(c_j)$, shifts the black line down (or up) by λ, except when $-\lambda \leq c_j \leq \lambda$, in which case it sets $w_j = 0$.

Optimality Conditions for Lasso

- On the contrast for hard thresholding, we set w_j to zero if $-\lambda \leq c_j \leq \lambda$, but it doesn't shrink the value of interval outside the interval.

- The slope of the soft thresholding line does not coincide with the diagonal, which means that even large coefficients are shrunk towards zero; consequently lasso is a biased estimator.

- This is undesirable, since if the likelihood indicates that the coefficient w_j should be large, we do not want to shrink it.

- Basically, lasso selects a subset of the variables, and shrinks all the coefficients by penalizing the absolute values.

  - Hence, the name "least absolute selection and shrinkage operator".

  - If $\lambda = 0$, we get OLS solution.

Optimality Conditions for Lasso

- If $\lambda \geq \lambda_{\max}$, we get $\widehat{\boldsymbol{w}} = 0$ where

$$\lambda_{\max} = \|\mathbf{X}^T \boldsymbol{y}\|_\infty = \max_j \left|\boldsymbol{y}^T \boldsymbol{x}_{:,j}\right|$$

- This value is computed using the fact that 0 is optimal if $(\mathbf{X}^T \boldsymbol{y})_j \in [-\lambda, \lambda], \forall j$.

- In general, the maximum penalty for an ℓ_1 regularized objective is

$$\lambda_{\max} = \max_j \left|\nabla_j \mathrm{NLL}(\mathbf{0})\right|$$

Comparison: Least Squares, Lasso, Ridge & Subset Selection

- We can gain further insight into ℓ_1 regularization by comparing it with least squares and ℓ_2 and ℓ_0 regularized least squares.

- For simplicity, assume all the features of $\mathbf{X}$ are orthonormal, so $\mathbf{X}^T\mathbf{X} = \mathbf{I}$. In this case,

$$\mathrm{RSS}(\boldsymbol{w}) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|^2 = \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{w}^T\mathbf{X}^T\mathbf{X}\boldsymbol{w} - 2\boldsymbol{w}^T\mathbf{X}^T\boldsymbol{y}$$
$$= \mathrm{const} + \sum_k w_k^2 - 2\sum_k\sum_i w_k x_{ik} y_i$$

- So we see this factorizes into a sum of terms, one per dimension. Hence we can write down the MAP and ML estimates analytically, as follows

  - **MLE:**

$$\widehat{\boldsymbol{w}}_k^{OLS} = \boldsymbol{x}_{:,k}^T\boldsymbol{y}$$

  - **Ridge:**

$$\widehat{\boldsymbol{w}}_k^{ridge} = \frac{\widehat{\boldsymbol{w}}_k^{OLS}}{1+\lambda}$$

Comparison: Least Squares, Lasso, Ridge & Subset Selection

- **LASSO:**

$$\widehat{\boldsymbol{w}}_k^{lasso} = \text{sign}\left(\widehat{\boldsymbol{w}}_k^{OLS}\right)\left(\left|\widehat{\boldsymbol{w}}_k^{OLS}\right| - \frac{\lambda}{2}\right)$$

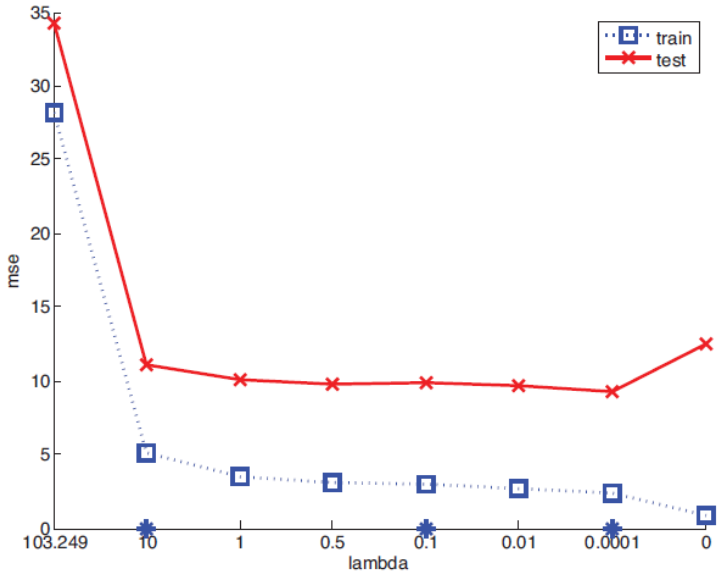
- **Subset selection:** If we pick the best K features using subset selection, the parameter estimate is as follows

$$\widehat{\boldsymbol{w}}_k^{SS} = \begin{cases} \widehat{\boldsymbol{w}}_k^{OLS} & \text{if rank}\left(\left|\boldsymbol{w}_k^{OLS}\right| \leq K\right) \\ 0 & \text{otherwise} \end{cases}$$

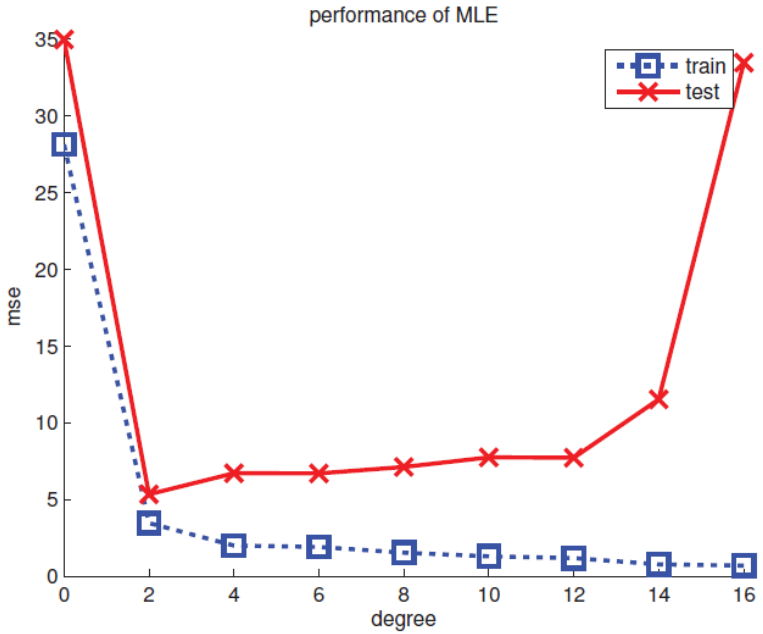
where rank refers to the location in the sorted list of weight magnitudes.

This corresponds to hard thresholding.

Comparison: Least Squares, Lasso, Ridge & Subset Selection



linRegPolyLassoDemo.m
from PMTK3



linRegPolyVsDegree.m
from PMTK3

(a) MSE vs λ for lasso for a degree 14 polynomial. Note that λ decreases as we move to the right.

(b) MSE versus polynomial degree. Note that the model order increases as we move to the right.

- Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer.

Comparison: Least Squares, Lasso, Ridge & Subset Selection

- The prostate cancer dataset considered has $D = 8$ features and $N = 67$ training cases; the goal is to predict the log prostate-specific antigen levels.

- Results of different methods on the prostate cancer data.

- Subset regression and lasso give sparse solutions. Bottom row is the mean squared error on the test set (30 cases)

Term	LS	Best Subset	Ridge	Lasso
Intercept	2.452	2.481	2.479	2.480
lcavol	0.716	0.651	0.656	0.653
lweight	0.293	0.380	0.300	0.297
age	-0.143	-0.000	-0.129	-0.119
lbph	0.212	-0.000	0.208	0.200
svi	0.310	-0.000	0.301	0.289
lcp	-0.289	-0.000	-0.260	-0.236
gleason	-0.021	-0.000	-0.019	0.000
pgg45	0.277	0.178	0.256	0.226
Test Error	0.586	0.572	0.580	0.564

- Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning*. Springer.

Regularization Path

Regularization Path

- Let us return to the most general case and write the function to be minimized as follows:

$$f(\boldsymbol{w}) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|^2 + 2\lambda\|\boldsymbol{w}\|_1 = \boldsymbol{y}^T\boldsymbol{y} + \boldsymbol{w}^T\mathbf{X}^T\mathbf{X}\boldsymbol{w} - 2\boldsymbol{w}^T\mathbf{X}^T\boldsymbol{y} + 2\lambda\|\boldsymbol{w}\|_1$$

- The subdifferential with w_j takes the form:

$$\partial(f)(\boldsymbol{w}) = \begin{cases} (2\mathbf{X}^T\mathbf{X}\boldsymbol{w} - 2\mathbf{X}^T\boldsymbol{y})_j + 2sign(w_j)\lambda \; if \; w_j \neq 0 \\ (2\mathbf{X}^T\mathbf{X}\boldsymbol{w} - 2\mathbf{X}^T\boldsymbol{y})_j + 2[-\lambda, \lambda] \qquad if \; w_j = 0 \end{cases}$$

- Once we find some $\widehat{\boldsymbol{w}}_\lambda$ such that $\partial(f)(\widehat{\boldsymbol{w}}_\lambda)$contains the zero vector a global minimum is assured.

- Consider $\lambda > \|\mathbf{X}^T\boldsymbol{y}\|_\infty = \max_j \left|\boldsymbol{y}^T\boldsymbol{x}_{:,j}\right|$. We have seen that $\widehat{\boldsymbol{w}}_\lambda = 0$ since the subdifferential contains the zero:

$$0 \in \partial(f)(\boldsymbol{w} = \mathbf{0}) = -2\boldsymbol{x}_j^T\boldsymbol{y} + 2[-\lambda, \lambda] \leftrightarrow \begin{matrix} -\boldsymbol{x}_j^T\boldsymbol{y} - \lambda < 0 \\ -\boldsymbol{x}_j^T\boldsymbol{y} + \lambda > 0 \end{matrix} \leftrightarrow \lambda > \left|\boldsymbol{y}^T\boldsymbol{x}_{:,j}\right|$$

Regularization Path

- Now, assume that the values of $\mathbf{X}^T\boldsymbol{y}$ are unique, positive, and ordered from highest to lower values. These conditions simply keep the notation easier.
- It turns out that for between $\boldsymbol{y}^T\boldsymbol{x}_{:,1}$ and $\boldsymbol{y}^T\boldsymbol{x}_{:,2}$ we obtain the same solution as in the uncorrelated case (with the addition of scaling factor as $\boldsymbol{x}_{:,1}^T\boldsymbol{x}_{:,1}$ which may not be 1):

$$\widehat{\boldsymbol{w}}_\lambda = \frac{1}{\boldsymbol{x}_{:,1}^T\boldsymbol{x}_{:,1}}\begin{pmatrix}\boldsymbol{y}^T\boldsymbol{x}_{:,1}-\lambda\\0\\\vdots\\0\end{pmatrix}$$

- Indeed for $j \neq 1$, we have:

$$0 \in -2\boldsymbol{x}_j^T\boldsymbol{y} + 2[-\lambda,\lambda] \leftrightarrow \begin{matrix}-\boldsymbol{x}_j^T\boldsymbol{y}-\lambda<0\\-\boldsymbol{x}_j^T\boldsymbol{y}+\lambda>0\end{matrix} \leftrightarrow \lambda > \left|\boldsymbol{y}^T\boldsymbol{x}_{:,j}\right|$$ (true since $\lambda > \left|\boldsymbol{y}^T\boldsymbol{x}_{:,1}\right|$ and $\boldsymbol{y}^T\boldsymbol{x}_{:,1}$ are ordered).

Regularization Path

$$\widehat{\boldsymbol{w}}_\lambda = \frac{1}{\boldsymbol{x}_{:,1}^T \boldsymbol{x}_{:,1}} \begin{pmatrix} \boldsymbol{y}^T \boldsymbol{x}_{:,1} - \lambda \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

- Similarly for $j = 1$, we have:

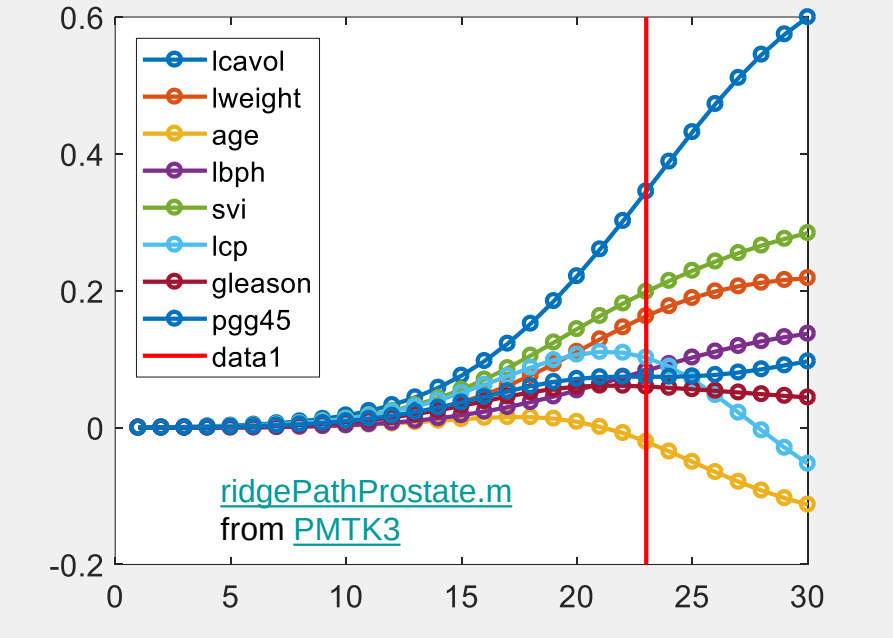
$$0 \in \left(2\boldsymbol{x}_{:,1}^T \boldsymbol{x}_{:,1} \boldsymbol{w} - 2\boldsymbol{x}_1^T \boldsymbol{y}\right)_j + 2sign(w_j)\lambda \leftrightarrow$$

$$0 \in 2\boldsymbol{x}_{:,1}^T \boldsymbol{x}_{:,1} w_1 - 2\boldsymbol{x}_1^T \boldsymbol{y} + 2\lambda \leftrightarrow w_1 = \frac{1}{\boldsymbol{x}_{:,1}^T \boldsymbol{x}_{:,1}} (\boldsymbol{y}^T \boldsymbol{x}_{:,1} - \lambda)$$

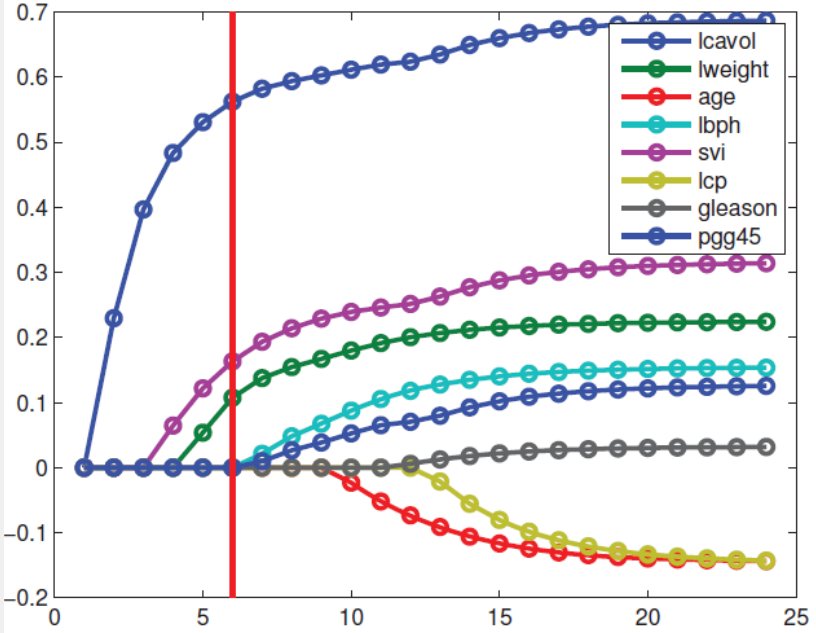
- For values with $\boldsymbol{x}_3^T \boldsymbol{y} \leq \lambda \leq \boldsymbol{x}_2^T \boldsymbol{y}$, the solution for $\widehat{\boldsymbol{w}}_\lambda$ is linear with non-zero elements in just the first 2 components.

Regularization Path

- As we increase λ, the solution vector $\widehat{\boldsymbol{w}}(\lambda)$ will tend to get sparser, although not necessarily monotonically.
- We can plot the values $\widehat{w}_j(\lambda)$ vs. λ for each feature j; this is known as the **regularization path**.



lassoPathProstate.m
from PMTK3

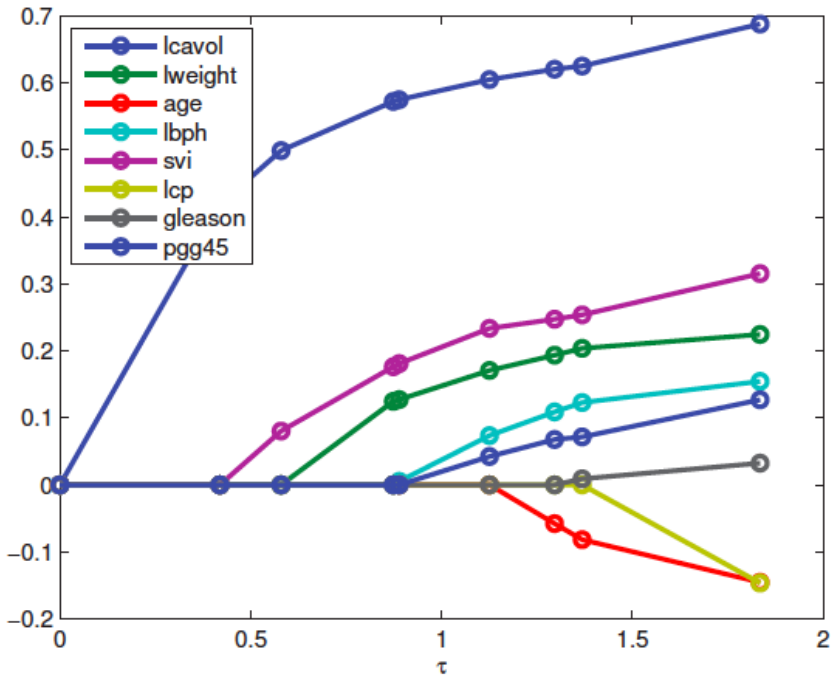


(a) Profiles of ridge coefficients for the prostate cancer example vs bound on ℓ_2 norm of $\boldsymbol{w}$, so small t (large λ) is on the left. The vertical line is the value chosen by 5-fold CV. (b) Profiles of lasso coefficients for the prostate cancer example vs bound on ℓ_1 norm of $\boldsymbol{w}$, so small t (large λ) is on the left.

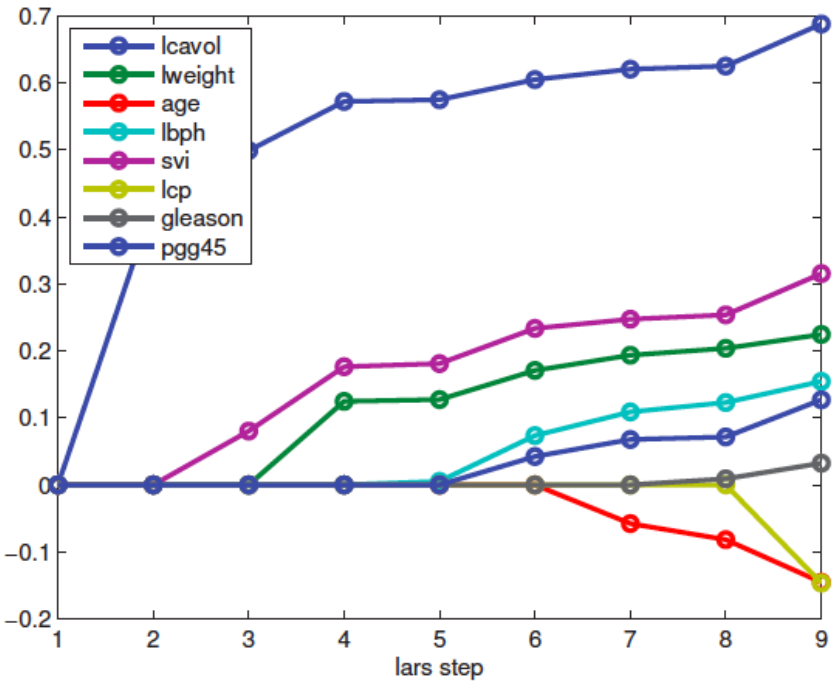
Regularization Path

- For ridge regression, we see that when $\lambda \to \infty$, all the coefficients go towards zero.
- But for any finite value of λ, all coefficients are non-zero.
- Similarly for LASSO, as we move to the right, the upper bound on the ℓ_1 penalty, B increases.
- When $B = 0$, all the coefficients are zero
- As we increase B, the coefficients gradually "turn on".
- But for any value between 0 and $B_{max} = \|\widehat{\boldsymbol{w}}_{OLS}\|_1$, the solution is sparse.
- Remarkably, it can be shown that the solution path is a piecewise linear function of B.
  - That is, there are a set of critical values of B where the active set of non-zero coefficients changes.
  - For values of B between these critical values, each non-zero coefficient increases or decreases in a linear fashion (see fig. in next slide)

Regularization Path



(a)



(b)

lassoPathProstate.m
from PMTK3

Illustration of piecewise linearity of regularization path for lasso on the prostate cancer example

(a) we plot $\widehat{w}_j$ vs B for the critical values of B.

(b) We plot vs steps of the LARS algorithm.

- Efron, B., I. Johnstone, T. Hastie, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*(2),407–499.

Regularization Path

- Furthermore, one can solve for these critical values analytically. This is the basis of the LARS algorithm.
- Remarkably, LARS can compute the entire regularization path for roughly the same computational cost as a single least squares fit.
  - By changing B from 0 to B_{max}, we can go from a solution in which all the weights are zero to a solution in which all weights are non-zero
- Unfortunately, not all subset sizes are achievable using lasso
- One can show that,
  - ✓ if $D > N$, the optimal solution can have at most N variables in it before reaching the complete set corresponding to the OLS solution of minimal ℓ_1 norm.
- One option to address this is the elastic nets (using simultaneously ℓ_1 and ℓ_2 regularization).

Model Selection

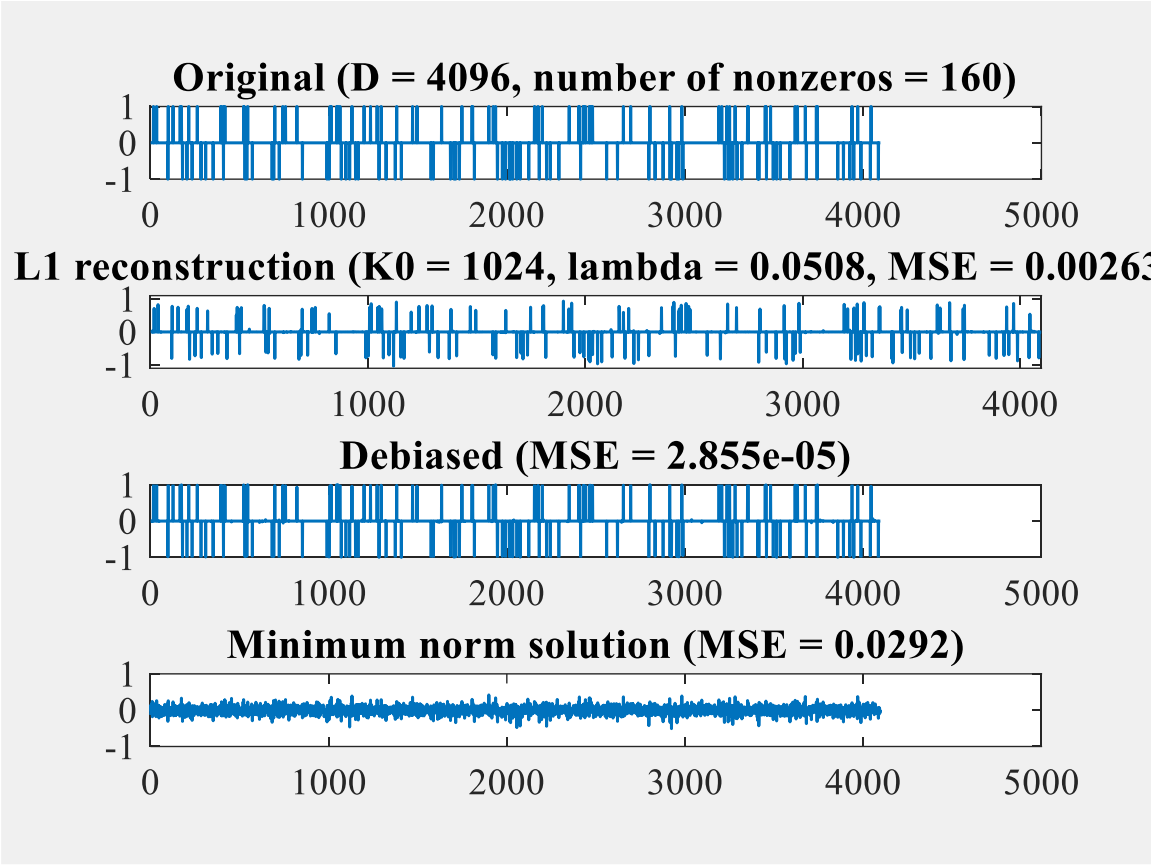
Model Selection

- It is tempting to use ℓ_1 regularization to estimate the set of relevant variables.
- In some cases, we can recover the true sparsity pattern of $\boldsymbol{w}^*$, the parameter vector that generated the data.
- A method that can recover the true model in the $N \rightarrow \infty$ limit is called model selection consistent.
- Consider and example in which we first generate a sparse signal w^* of size $D = 4096$, consisting of 160 randomly placed ± 1 spikes.
- Next we generate a random design matrix $\mathbf{X}$ of size $N \times D$, where $N = 1024$.
- Finally we generate a noisy observation $\boldsymbol{y} = \mathbf{X}\boldsymbol{w}^* + \boldsymbol{\epsilon}$ where $\epsilon \sim \mathcal{N}(0, 0.001^2)$. We then estimate $\boldsymbol{w}$.

- Buhlmann, P. and S. van de Geer (2011). *Statistics for High- Dimensional Data: Methodology, Theory and Applications*. Springer.

Model Selection

- Example of recovering a sparse signal using lasso
- The original $\boldsymbol{w}^*$ is shown in the first row.
- The second row is the ℓ_1 estimate $\widehat{\boldsymbol{w}}_{L1}$ using $\lambda = 0.1\lambda_{max}$. We see that this has "spikes" in the right places, but they are too small.
- The third row is the least squares estimate of the coefficients which are estimated to be non-zero based on $supp(\widehat{\boldsymbol{w}}_{L1})$.



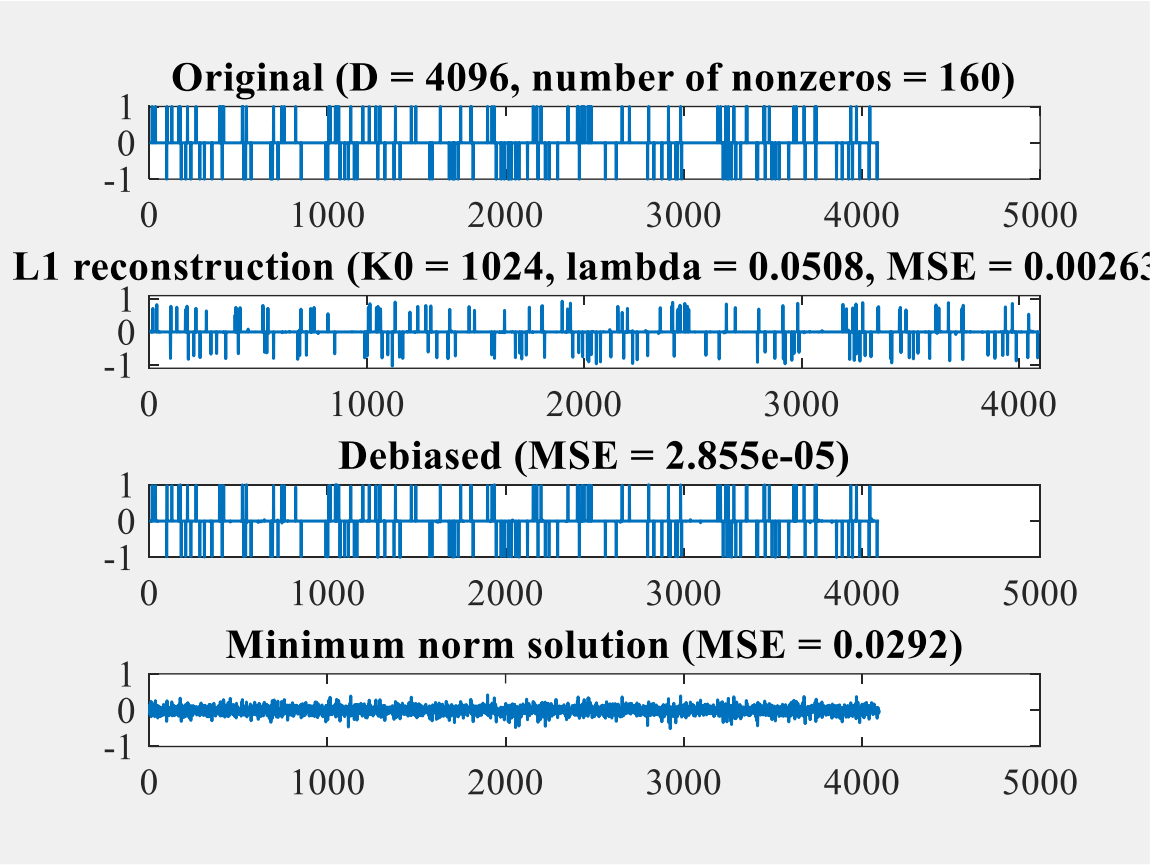
sparseSensingDemo.m
from PMTK3

- Figueiredo, M., R. Nowak, and S. Wright (2007). Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE. J. on Selected Topics in Signal Processing*.

Model Selection

- The third row is the LS estimate of the coefficients which are estimated to be non-zero based on $supp(\widehat{\boldsymbol{w}}_{L1})$. This is called **debiasing**, and is necessary because lasso shrinks the relevant coefficients as well as the irrelevant ones.

- The last row is the least squares estimate for all the coefficients jointly, ignoring sparsity.

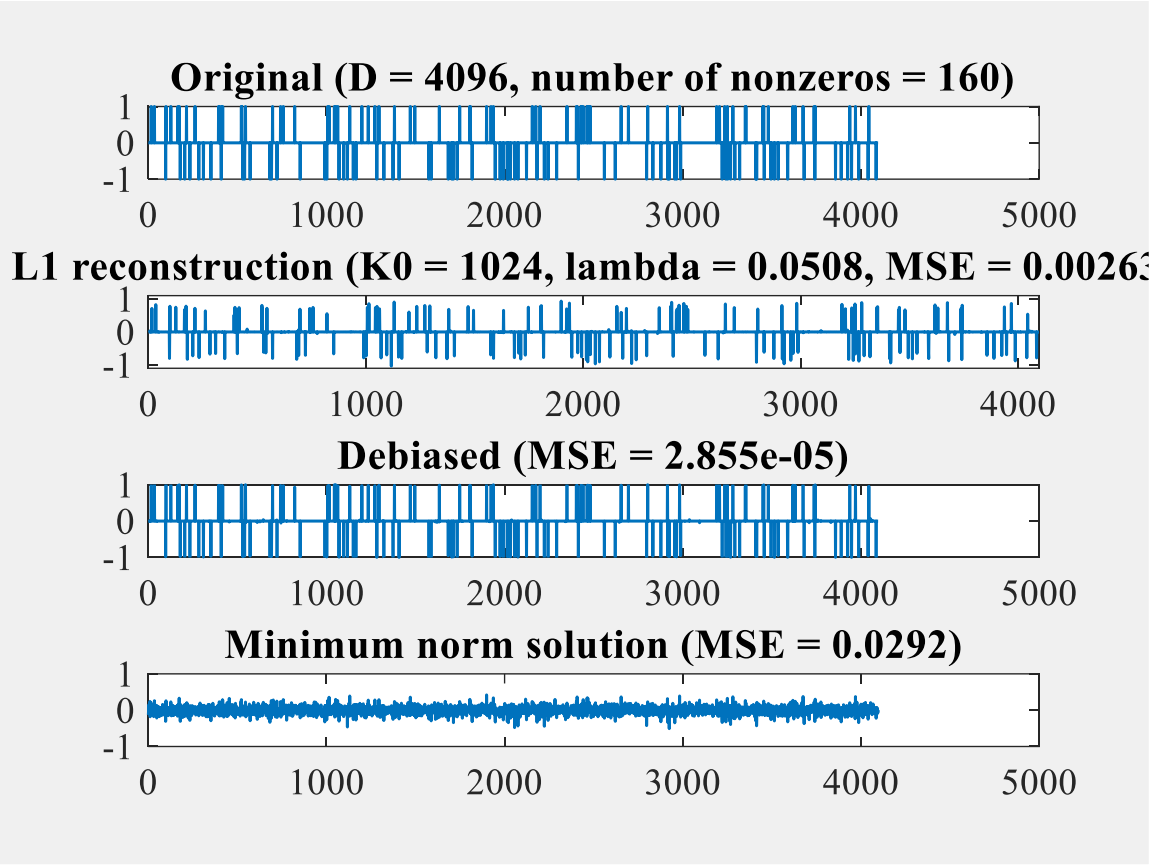


sparseSensingDemo.m
from PMTK3

- Figueiredo, M., R. Nowak, and S. Wright (2007). Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE. J. on Selected Topics in Signal Processing*.

Model Selection

- The (debiased) sparse estimate is an excellent estimate of the original signal.
- By contrast, least squares without the sparsity assumption performs very poorly.



sparseSensingDemo.m
from PMTK3

- Figueiredo, M., R. Nowak, and S. Wright (2007). Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE. J. on Selected Topics in Signal Processing*.

Model Selection

- To perform model selection, we have to pick λ.
- It is common to use cross validation.
- However, it is important to note that cross validation is picking a value of λ that results in good predictive accuracy.
- This is not usually the same value as the one that is likely to recover the "true" model.
- To see why, recall that ℓ_1 regularization performs selection and shrinkage, that is, the chosen coefficients are brought closer to 0.
- In order to prevent relevant coefficients from being shrunk in this way, cross validation will tend to pick a value of λ that is not too large.
- This will result in a less sparse model which contains irrelevant variables (false positives).

- Meinshausen, N. and P. Bãijhlmann (2010). Stability selection. *J. of Royal Stat. Soc. Series B 72*, 417–473.

Model Selection

- A downside of using ℓ_1 regularization to select variables is that it can give quite different results if the data is perturbed slightly.
- The Bayesian approach, which estimates posterior marginal inclusion probabilities, $p(\gamma_j = 1|\mathcal{D})$, is much more robust.
- A frequentist solution to this is to use bootstrap resampling, and to rerun the estimator on different versions of the data.
- By computing how often each variable is selected across different trials, we can approximate the posterior inclusion probabilities.
- This method is known as stability selection.

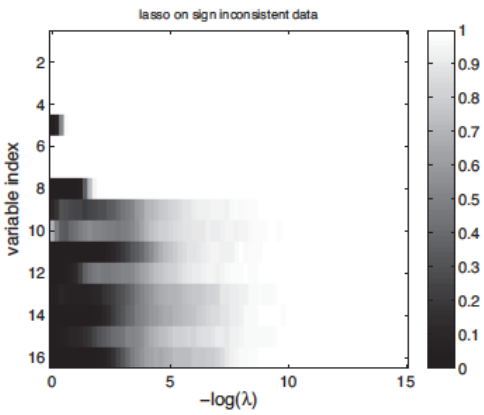
- Meinshausen, N. and P. Bãijhlmann (2010). Stability selection. *J. of Royal Stat. Soc. Series B 72*, 417–473.

Bootstrap Lasso (Bolasso)

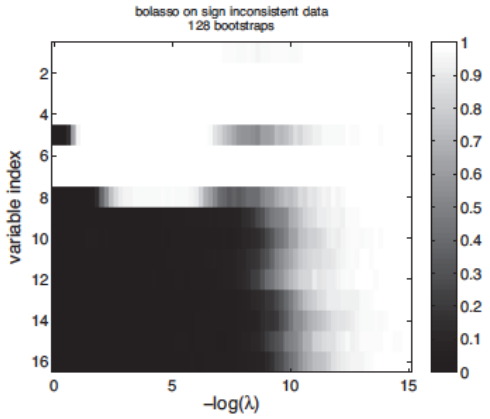
- We can threshold the stability selection (bootstrap) inclusion probabilities at some level, say 90%, and thus derive a sparse estimator.
- This is known as bootstrap lasso or bolasso.
- It will include a variable if it occurs in at least 90% of sets returned by lasso.
- This process of intersecting the sets is a way of eliminating the false positives that vanilla lasso produces.
- The theoretical results in (Bach 2008) prove that bolasso is model selection consistent under a wider range of conditions than vanilla lasso.
- As an illustration, we reproduced the experiments in (Bach 2008)
  - In particular, we created 256 datasets of size $N = 1000$ with $D = 16$ variables, of which 8 are relevant.

- Bach, F. (2008). Bolasso: Model Consistent Lasso Estimation through the Bootstrap. In *Intl. Conf. on Machine Learning*.

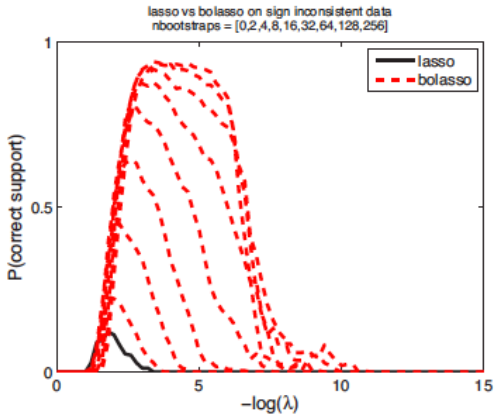
Bolasso



(a)



(b)



(c)

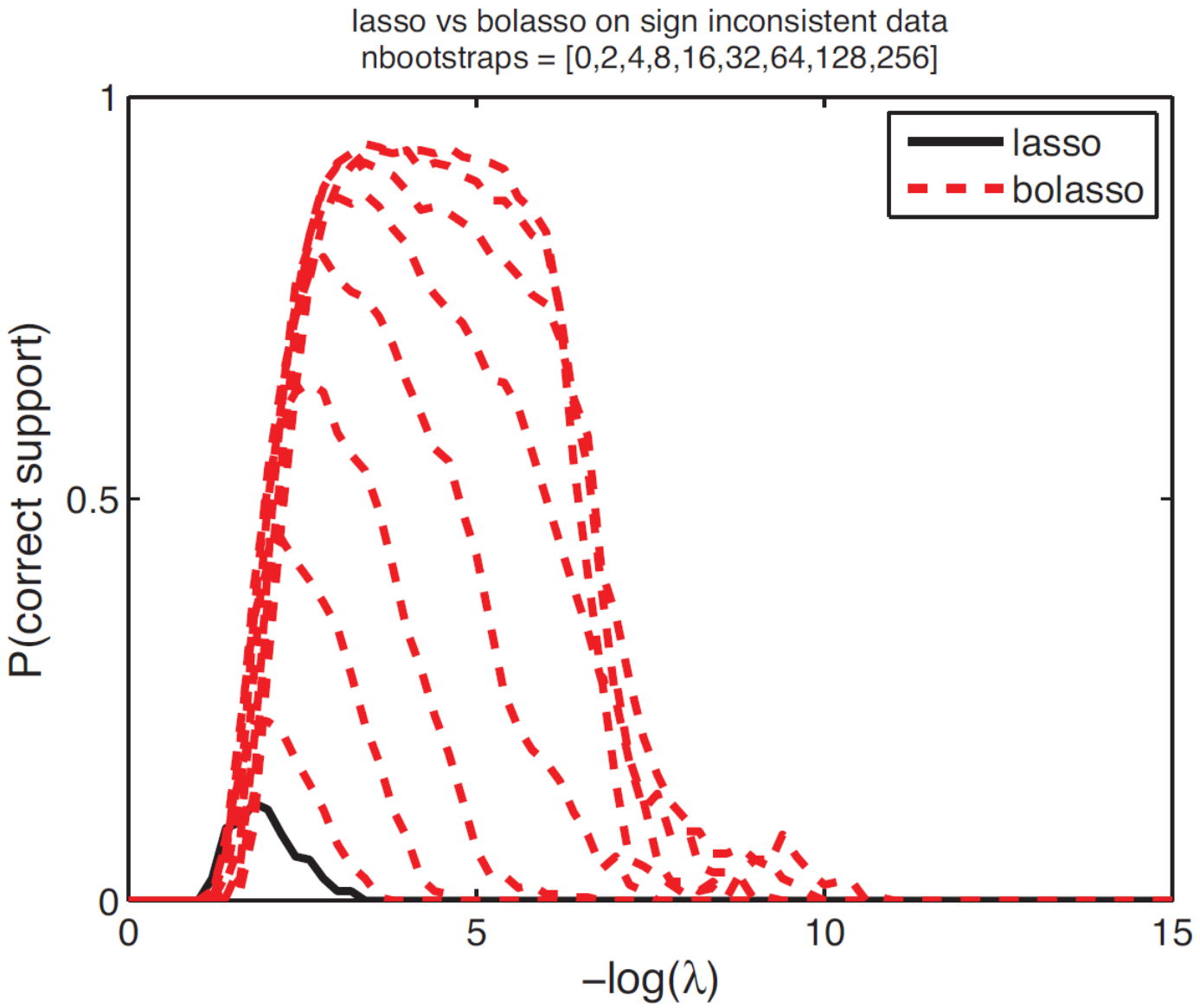
BolassoDemo.m
from PMTK3

(a) Probability of selection of each variable (white = large probabilities, black = small probabilities) vs. regularization parameter for Lasso. As we move from left to right, we decrease the amount of regularization, and therefore select more variables.

(b) Same as (a) but for bolasso.

(c) Probability of correct sign estimation vs. regularization parameter. Bolasso (red, dashed) and Lasso (black, plain): The number of bootstrap replications is in $\{2, 4, 8, 16, 32, 64, 128, 256\}$.

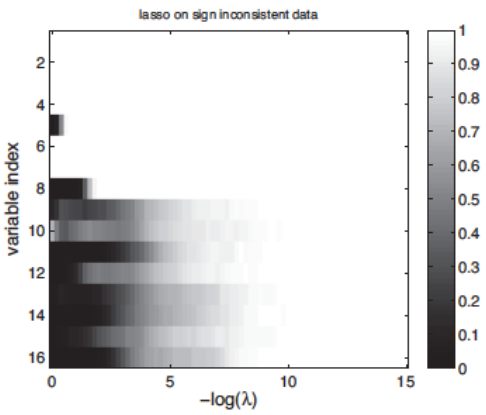
Bolasso



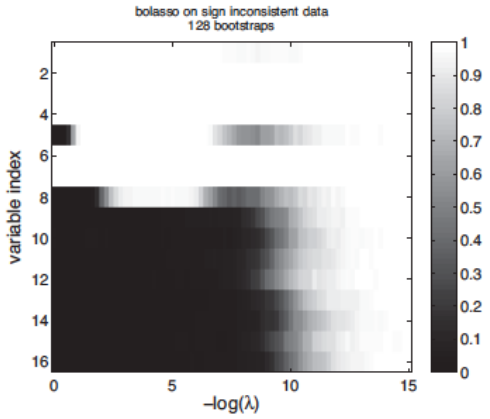
Probability of correct sign estimation vs. regularization parameter.

Bolasso (red, dashed) and Lasso (black, plain): The number of bootstrap replications is in $\{2, 4, 8, 16, 32, 64, 128, 256\}$.

Bolasso

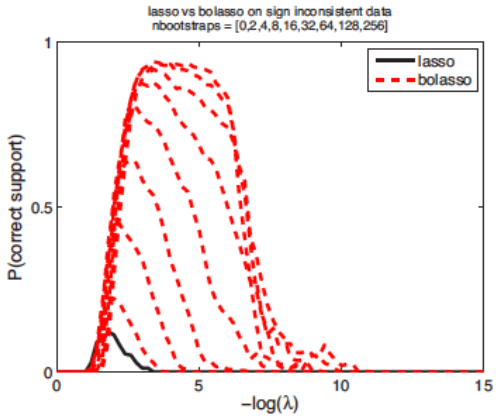


(a)



(b)

BolassoDemo.m
from PMTK3

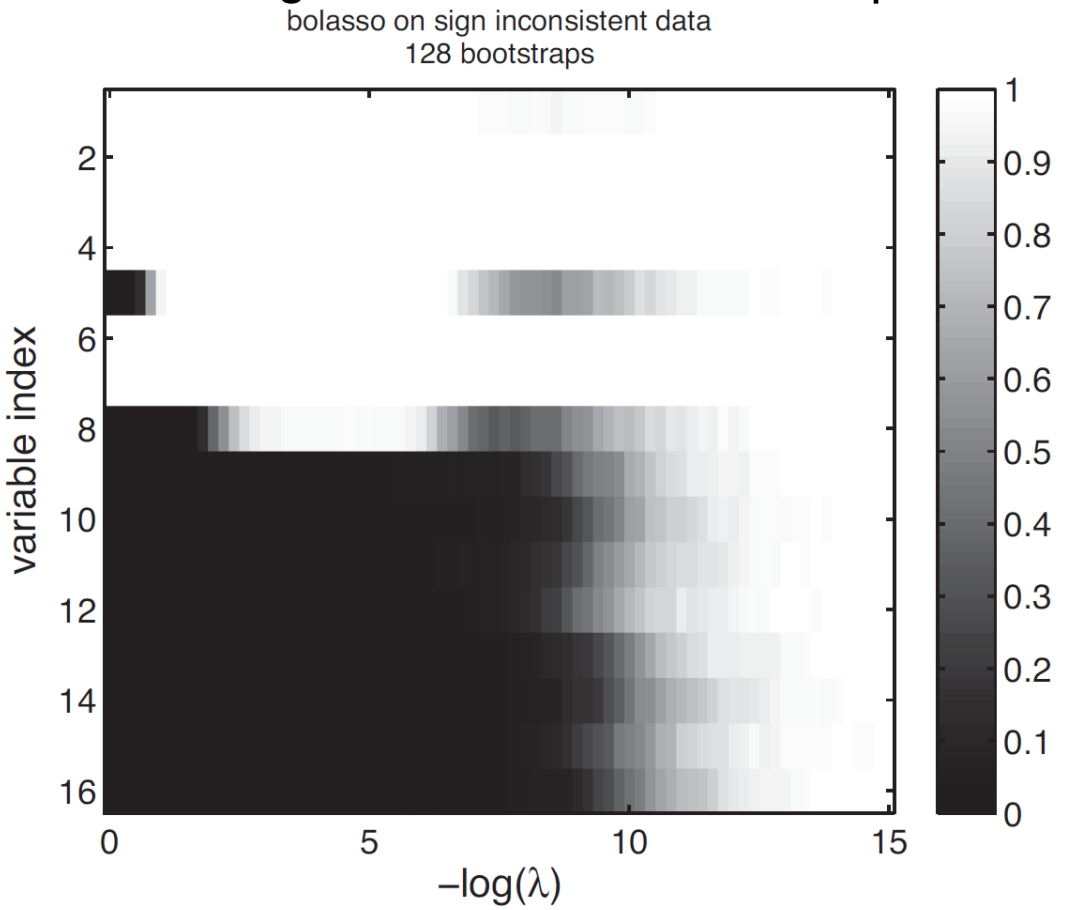


(c)

- For dataset n, variable j, and sparsity level k, define $S(j,k,n) = \mathbb{I}(\widehat{w}_j(\lambda_k, \mathcal{D}_n) \neq 0)$. Now define $P(j,k)$ be the average of $S(j,k,n)$ over the 256 datasets.
- In Figs (a-b), we plot P vs $-\log(\lambda)$ for lasso and bolasso. We see that for bolasso, there is a large range of λ where the true variables are selected, but this is not the case for lasso.
- In Fig. (c) we plot the empirical probability that the correct set of variables is recovered, for lasso and for bolasso with an increasing number of bootstrap samples. In practice, 32 bootstraps seems to be a good compromise between speed and accuracy.

Bolasso

- With bolasso, there is the usual issue of picking λ.
- Obviously we could use cross validation, but there exists another heuristic: shuffle the rows to create a large black block, and then pick λ to be in the middle of this region.
- A Bayesian approach provides a more principled method for selecting λ.



Bayesian inference for Linear Models with Laplace Priors

- We have been focusing on MAP estimation in sparse linear models.
- It is also possible to perform Bayesian inference.
- However, the posterior mean and median, as well as samples from the posterior, are not sparse; only the mode is sparse.
- Another argument in favor of using the posterior mean comes from the fact that plugging in the posterior mean, rather than the posterior mode, is the optimal thing to do if we want to minimize squared prediction error.
- Using the posterior mean with a spike-and-slab prior results in better prediction accuracy.

- Park, T. and G. Casella (2008). The Bayesian Lasso. *J. of the Am. Stat. Assoc. 103*(482), 681–686.
- Seeger, M. (2008). Bayesian Inference and Optimal Design in the Sparse Linear Model. *J. of Machine Learning Research 9*, 759–813.
- Elad, A plurality of sparse representations is better than the M. and I. Yavnch (2009). sparsest one alone. *IEEE Trans. on Info. Theory 55*(10), 4701–4714.
- Schniter, P., L. C. Potter, and J. Ziniel (2008). Fast Bayesian Matching Pursuit: Model Uncertainty and Parameter Estimation for Sparse Linear Models. Technical report, U. Ohio. Submitted to IEEE Trans. On Signal Processing.

ℓ_1 Regularization Algorithms

ℓ_1 Regularization Algorithms – Coordinate Descent

- Sometimes it is hard to optimize all the variables simultaneously, but it easy to optimize them one by one.
- In particular, we can solve for the j-th coefficient with all the others held fixed

$$w_j^* = \arg\min_z f\left(\boldsymbol{w} + z\boldsymbol{e}_j\right) - f(\boldsymbol{w})$$

where $\boldsymbol{e}_j$ is the j –th unit vector.

- We can either cycle through the coordinates in a deterministic fashion, or we can sample them at random, or we can choose to update the coordinate for which the gradient is steepest.
- The coordinate descent method is particularly appealing if each one-dimensional optimization problem can be solved analytically.

- Yaun, G.-X., K.-W. Chang, C.-J. Hsieh, and C.-J. Lin (2010). A Comparison of Optimization Methods and Software for Large-scale L1-regularized Linear Classification. *J. of Machine Learning Research 11*, 3183–3234.
- Yang, A., A. Ganesh, S. Sastry, and Y. Ma (2010, Feb). Fast l1- minimization algorithms and an application in robust face recognition: A review. Technical Report UCB/EECS-2010-13, EECS Department, University of California, Berkeley.
- Schmidt, M., G. Fung, and R. Rosales (2009). Optimization methods for ℓ − 1 regularization. Technical report, U. British Columbia.
- Fu, W. (1998). Penalized regressions: the bridge verus the lasso. *J. Computational and graphical statistics*.
- Wu, T. T. and K. Lange (2008). Coordinate descent algorithms for lasso penalized regression. *Ann. Appl. Stat 2*(1), 224–244.

ℓ_1 Regularization Algorithms - Coordinate Descent

Coordinate descent for lasso (aka shooting algorithm)

1 Initialize $\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$;
2 **repeat**
3 **for** $j = 1, \ldots, D$ **do**
4 $a_j = 2\sum_{i=1}^{n} x_{ij}^2$;
5 $c_j = 2\sum_{i=1}^{n} x_{ij}(y_i - \mathbf{w}^T\mathbf{x}_i + w_j x_{ij})$;
6 $w_j = \text{soft}(\frac{c_j}{a_j}, \frac{\lambda}{a_j})$;
7 **until** *converged*;

Active Set Methods

- The problem with coordinate descent is that it only updates one variable at a time, so can be slow to converge.
- Active set methods update many variables at a time.
- Unfortunately, they are more complicated, because of the need to identify which variables are constrained to be zero, and which are free to be updated.
- Active set methods typically only add or remove a few variables at a time, so they can take a long if they are started far from the solution.
- But they are ideally suited for generating a set of solutions for different values of λ starting with the empty set, i.e., for generating regularization path.
- These algorithms exploit the fact that one can quickly compute $\widehat{\boldsymbol{w}}(\lambda_k)$ from $\widehat{\boldsymbol{w}}(\lambda_{k-1})$ if $\lambda_k \approx \lambda_{k-1}$.

Least Angle Regression and Shrinkage (LARS)

- In fact, even if we only want the solution for a single value of λ, say λ_* it can sometimes be computationally more efficient to compute a set of solutions, from λ_{max} to λ_* using warm-starting.
  - This is called a **continuation method** or **homotopy method**.
- This is often much faster than directly "cold-starting" at λ_*; this is particularly true if λ_* is small.
- Perhaps the most well-known example of a homotopy method in machine learning is the LARS algorithm, which stands for "least angle regression and shrinkage".
- This can compute $\widehat{\boldsymbol{w}}(\lambda)$ for all possible values of λ in an efficient manner.

- Efron, B., I. Johnstone, T. Hastie, and R. Tibshirani (2004). Least angle regression. *Annals of Statistics 32*(2),407–499.
- Osborne, M. R., B. Presnell, and B. A. Turlach (2000a). A new approach to variable selection in least squares problems. *IMA Journal of Numerical Analysis 20*(3), 389–403.
- Osborne, M. R., B. Presnell, and B. A. Turlach (2000b). On the lasso and its dual. *J. Computational and graphical statistics 9*, 319–337.

Least Angle Regression and Shrinkage (LARS)

- LARS works as follows
  - It starts with a large value of λ, such that only the variable that is most correlated with the response vector $\boldsymbol{y}$ is chosen.
  - Then, λ is decreased until a second variable is found which has the same correlation (in terms of magnitude) with the current residual as the first variable.
    - Residual at step k is defined as $r_k = \boldsymbol{y} - \mathbf{X}_{:,F_k}\boldsymbol{w}_k$ where, F_k is the current active set.
    - Remarkably, one can solve for this new value of λ analytically by using a geometric argument.
    - This allows the algorithm to quickly "jump" to the next point on the regularization path where the active set changes. This repeats until all the variables are added.
- It is necessary to allow variables to be removed from the active set if we want the sequence of solutions to correspond to the regularization path of lasso.
  - In LARS, we disallow variable removal and hence, it is faster.

LARS and other Homotopy Methods

- In particular, LARS costs the same as a single ordinary least squares fit, namely $\mathcal{O}(ND\min(N,D))$.
- LARS is very similar to greedy forward selection, and a method known as least squares boosting.
- There have been many attempts to extend the LARS algorithm to compute the full regularization path for ℓ_1 regularized GLMs, such as logistic regression.
- In general, one cannot analytically solve for the critical values of λ.
- The method described in (Friedman et al. 2010) combines coordinate descent with this warm-starting strategy, and computes the full regularization path for any ℓ_1 regularized GLM.

- Friedman, J., T. Hastie, and R. Tibshirani (2010, Februrary). Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. of Statistical Software 33*(1).

Proximal and Gradient Projection Methods

- Consider a convex objective of the form

$$f(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + R(\boldsymbol{\theta})$$

- $L(\boldsymbol{\theta})$ (representing the loss) is convex and differentiable.

- $R(\boldsymbol{\theta})$ (representing the regularizer) is convex but no necessarily differentiable.

- The lasso problem can be formulated as follows:

$$L(\boldsymbol{\theta}) = \text{RSS}(\boldsymbol{\theta}), \qquad R(\boldsymbol{\theta}) = I_C(\boldsymbol{\theta})$$

where

$$C = \{\boldsymbol{\theta}: \|\boldsymbol{\theta}\|_1 \leq B\}$$

and

$$I_C(\boldsymbol{\theta}) \triangleq \begin{cases} 0, & \boldsymbol{\theta} \in C \\ +\infty, & \text{otherwise} \end{cases}$$

- Yang, A., A. Ganesh, S. Sastry, and Y. Ma (2010, Feb). Fast l1- minimization algorithms and an application in robust face recognition: A review. Technical Report UCB/EECS-2010-13, EECS Department, University of California, Berkeley.

Proximal and Gradient Projection Methods

- In some cases, it is easy to optimize functions of the form $f(\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + R(\boldsymbol{\theta})$.
- For example, suppose $L(\boldsymbol{\theta}) = \text{RSS}(\boldsymbol{\theta})$, and the design matrix is simply $\mathbf{X} = \mathbf{I}$.
- Then the objective becomes

$$f(\boldsymbol{\theta}) = R(\boldsymbol{\theta}) + \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{y}\|_2^2$$

- The minimizer of this is given by $\text{prox}_R(\boldsymbol{y})$ which is the proximal operator for the convex function R, defined by

$$\text{prox}_R(\boldsymbol{y}) = \arg\min_{\boldsymbol{z}} \left(R(\boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{y}\|_2^2 \right)$$

- Intuitively, we are returning a point that minimizes R but which is also close (proximal) to y.
- In general, we will use this operator inside an iterative optimizer, in which case we want to stay close to the previous iterate.

- Vandenberghe, L. (2011). Ee236c – optimization methods for large-scale systems.
- Yang, A., A. Ganesh, S. Sastry, and Y. Ma (2010, Feb). Fast l1- minimization algorithms and an application in robust face recognition: A review. Technical Report UCB/EECS-2010-13, EECS Department, University of California, Berkeley.

Proximal Operators

- In this case, we use

$$\text{prox}_R(\boldsymbol{\theta}_k) = \arg\min_{\boldsymbol{z}} \left(R(\boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{\theta}_k\|_2^2 \right)$$

- The key issues are:
  - how do we efficiently compute the proximal operator for different regularizers R
  - how do we extend this technique to more general loss functions L?
- If $R(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_1$, the proximal operator is given by component-wise soft-thresholding

$$\text{prox}_R(\boldsymbol{\theta}) = \text{soft}(\boldsymbol{\theta}, \lambda)$$

- If $R(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_0$, the proximal operator is given by component wise hard-thresholding

$$\text{prox}_R(\boldsymbol{\theta}) = \text{hard}\left(\boldsymbol{\theta}, \sqrt{2\lambda}\right)$$

where $\text{hard}(u, a) \triangleq u\mathbb{I}(|u| > a)$.

Proximal Operators

- If $R(\boldsymbol{\theta}) = I_C(\boldsymbol{\theta})$, the proximal operator is given by the projection onto the set C:

$$\text{prox}_R(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{z} \in C} \|\boldsymbol{z} - \boldsymbol{\theta}\|_2^2 = \text{proj}_C(\boldsymbol{\theta})$$

- For some convex sets, it is easy to compute the projection operator.

- For example, to project onto the rectangular set defined by the box constraints $C = \{\boldsymbol{\theta}: \ell_j \leq \theta_j \leq u_j\}$, we can use

$$\text{proj}_C(\boldsymbol{\theta})_j = \begin{cases} \ell_j, & \theta_j \leq \ell_j \\ \theta_j, & \ell_j \leq \theta_j \leq u_j \\ u_j, & \theta_j \geq u_j \end{cases}$$

- To project onto the Euclidean ball $C = \{\boldsymbol{\theta}: \|\boldsymbol{\theta}\|_2 \leq 1\}$ we can use

$$\text{proj}_C(\boldsymbol{\theta}) = \begin{cases} \boldsymbol{\theta}/\|\boldsymbol{\theta}\|_2, & \|\boldsymbol{\theta}\|_2 > 1 \\ \boldsymbol{\theta}, & \|\boldsymbol{\theta}\|_2 \leq 1 \end{cases}$$

Proximal and Gradient Projection Methods

- To project onto the 1-norm ball $C = \{\boldsymbol{\theta}: \|\boldsymbol{\theta}\|_1 \leq 1\}$ we can use

$$\text{proj}_C(\boldsymbol{\theta}) = \text{soft}(\boldsymbol{\theta}, \lambda)$$

where $\lambda = 0$ if $\|\boldsymbol{\theta}\|_1 \leq 1$, and otherwise λ is the solution to the equation

$$\sum_{j=1}^{D} \max\left(\left|\theta_j\right| - \lambda, 0\right) = 1$$

- We can implement the whole procedure in $\mathcal{O}(D)$ time, as explained in (Duchi et al. 2008).

- Duchi, J., S. Shalev-Shwartz, Y. Singer, and T. Chandra (2008). Efficient projections onto the L1-ball for learning in high dimensions. In *Intl. Conf. on Machine Learning*.

Proximal Gradient Method

- To use the proximal operator inside of a gradient descent routine, the basic idea is to minimize a simple quadratic approximation to the loss function centered on the $\boldsymbol{\theta}_k$

$$\boldsymbol{\theta}_{k+1} = \arg\min_{z} R(\boldsymbol{z}) + L(\boldsymbol{\theta}_k) + g_k^T(\boldsymbol{z} - \boldsymbol{\theta}_k) + \frac{1}{2t_k}\|\boldsymbol{z} - \boldsymbol{\theta}_k\|_2^2$$

- $g_k = \nabla L(\theta_k)$ is the gradient of the loss, t_k is a constant.

- The last term arises from a simple approximation to the Hessian of the loss of the form $\nabla^2 L(\theta_k) \approx \frac{1}{t_k}\mathbf{I}$.

- Duchi, J., S. Shalev-Shwartz, Y. Singer, and T. Chandra (2008). Efficient projections onto the L1-ball for learning in high dimensions. In *Intl. Conf. on Machine Learning*.

Proximal Gradient Method

$$\boldsymbol{\theta}_{k+1} = \arg\min_{z} R(\boldsymbol{z}) + L(\boldsymbol{\theta}_k) + g_k^T(\boldsymbol{z} - \boldsymbol{\theta}_k) + \frac{1}{2t_k}\|\boldsymbol{z} - \boldsymbol{\theta}_k\|_2^2$$

- Dropping terms that are independent of $\boldsymbol{z}$, and multiplying t_k we can rewrite the above expression in terms of a proximal operator as follows

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{z}} \left[t_k R(\boldsymbol{z}) + \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{u}_k\|_2^2 \right] = \text{prox}_{t_k R}(\boldsymbol{u}_k),$$

$$\boldsymbol{u}_k = \boldsymbol{\theta}_k - t_k \boldsymbol{g}_k,$$

$$\boldsymbol{g}_k = \nabla L(\boldsymbol{\theta}_k).$$

- If $R(\boldsymbol{\theta}) = 0$, this is equivalent to gradient descent.
- If $R(\boldsymbol{\theta}) = I_C(\boldsymbol{\theta})$, the method is equivalent to projected gradient descent.
- If $R(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_1$, the method is equivalent to iterative gradient descent.

Projected Gradient Descent

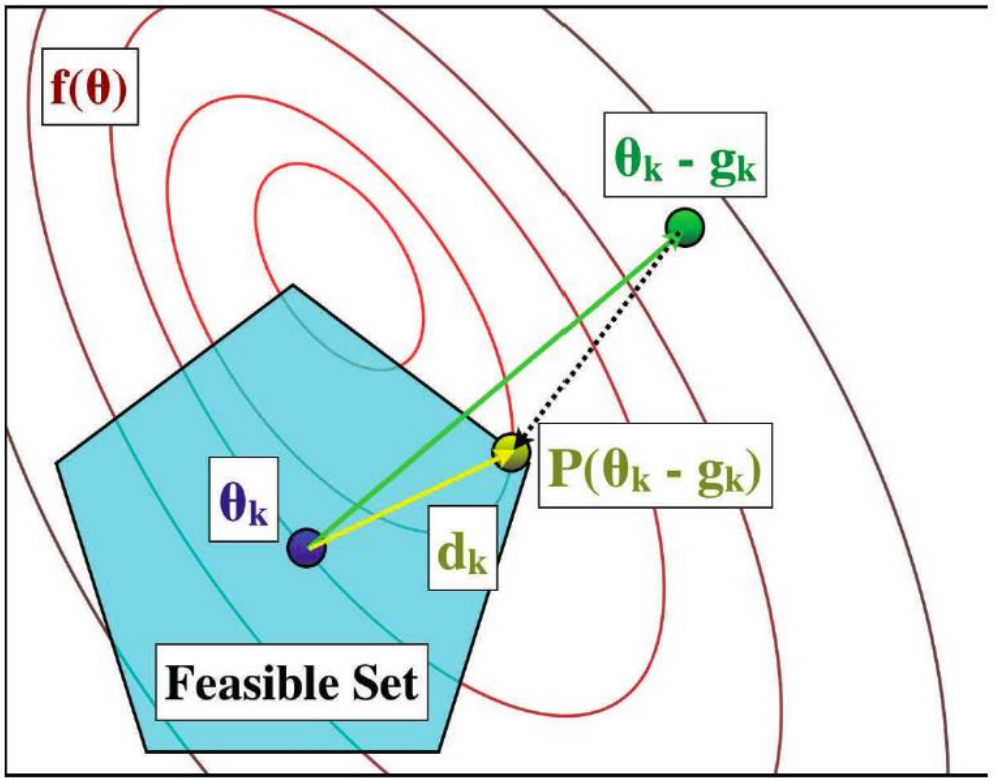
- Illustration of projected gradient descent. The step along the negative gradient, to $\boldsymbol{\theta}_k - \boldsymbol{g}_k$, takes us outside the feasible set.

- If we project that point onto the closest point in the set we get

$$\boldsymbol{\theta}_{k+1} = proj_{\boldsymbol{\Theta}}(\boldsymbol{\theta}_k - \boldsymbol{g}_k).$$

- We can then derive the implicit update direction using

$$\boldsymbol{d}_k = \boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k$$



Barzilai-Borwein Step-Size

- There are several ways to pick t_k, or equivalently $\alpha_k = 1/t_k$.
- Since $\alpha_k \mathbf{I}$ is an approximation to the Hessian $\nabla^2 L$, we require that

$$\alpha_k(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) \approx g_k - g_{k-1}$$

  in the least squares sense.

- Therefore,

$$\alpha_k = \arg\min_{\alpha} \|\alpha(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1}) - (\boldsymbol{g}_k - \boldsymbol{g}_{k-1})\|_2^2$$

$$= \frac{(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})^T(\boldsymbol{g}_k - \boldsymbol{g}_{k-1})}{(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})^T(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})}$$

- This is known as the Barzilai-Borwein (BB) or spectral step-size.
- This step-size can be used with any gradient method, whether proximal or not.

- Barzilai, J. and J. Borwein (1988). Two point step size gradient methods. *IMA J. of Numerical Analysis 8*, 141– 148.
- Fletcher, R. (2005). On the Barzilai- Borwein Method. *Applied Optimization 96*, 235–256.
- Raydan, M. (1997). The barzilai and borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. on Optimization 7* (1), 26–33.

Iterative Shrinkage-Thresholding

- The BB step size does not lead to monotonic decrease of the objective but it is much faster than standard line search techniques.
- To ensure convergence, we require that the objective decrease "on average", where the average is computed over a sliding window of size $M + 1$.
- When we combine
  - ✓ the BB stepsize with
  - ✓ the iterative soft thresholding technique (for $R(\boldsymbol{\theta}) = \lambda\|\boldsymbol{\theta}\|_1$), plus
  - ✓ a continuation method that gradually reduces λ,

  we get a fast method for the **basis pursuit denoising (**BPDN) problem known as the SpaRSA algorithm, which stands for "sparse reconstruction by separable approximation".
- However, we will call it the iterative shrinkage and thresholding algorithm.

- Wright, S., R. Nowak, and M. Figueiredo (2009). Sparse reconstruction by separable approximation. *IEEE Trans. on Signal Processing 57* (7), 2479–2493.

Iterative Shrinkage-Thresholding

Iterative Shrinkage-Thresholding Algorithm (ISTA)

Input: $\mathbf{X} \in \mathbb{R}^{N \times D}$, $\mathbf{y} \in \mathbb{R}^{N}$, parameters $\lambda \geq 0$, $M \geq 1$, $0 < s < 1$;
Initialize $\boldsymbol{\theta}_0 = \mathbf{0}$, $\alpha = 1$, $\mathbf{r} = \mathbf{y}$, $\lambda_0 = \infty$;
repeat
  $\lambda_t = \max(s||\mathbf{X}^T\mathbf{r}||_\infty, \lambda)$ // Adapt the regularizer ;
  repeat
    $\mathbf{g} = \nabla L(\boldsymbol{\theta})$;
    $\mathbf{u} = \boldsymbol{\theta} - \frac{1}{\alpha}\mathbf{g}$;
    $\boldsymbol{\theta} = \text{soft}(\mathbf{u}, \frac{\lambda_t}{\alpha})$;
    Update α using BB stepsize in Equation
  until *$f(\boldsymbol{\theta})$ increased too much within the past M steps*;
  $\mathbf{r} = \mathbf{y} - \mathbf{X}\boldsymbol{\theta}$ // Update residual ;
until $\lambda_t = \lambda$;

$$\alpha_k = \frac{(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})^T(\boldsymbol{g}_k - \boldsymbol{g}_{k-1})}{(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})^T(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})}$$

Nesterov's Method

- A faster version of proximal gradient descent can be obtained by expanding the quadratic approximation around a point other than the most recent parameter value.
- In particular, consider performing updates of the form

$$\theta_{k+1} = \text{prox}_{t_k R}(\boldsymbol{\phi}_k - \boldsymbol{t}_k \boldsymbol{g}_k), \qquad \boldsymbol{g}_k = \nabla L(\phi_k)$$

$$\boldsymbol{\phi}_k = \boldsymbol{\theta}_k + \frac{k-1}{k+2}(\boldsymbol{\theta}_k - \boldsymbol{\theta}_{k-1})$$

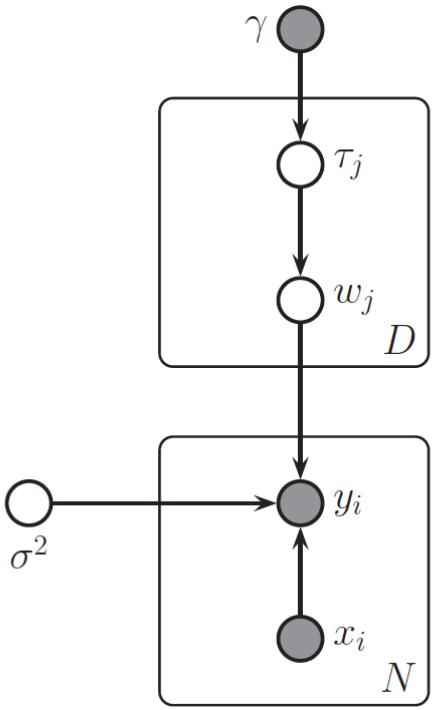
- This is known as Nesterov's method.
- As before, there are a variety of ways of setting $\boldsymbol{t}_k$; typically one uses line search.
- When combined with the iterative soft thresholding technique (for $R(\boldsymbol{\theta}) = \lambda \|\boldsymbol{\theta}\|_1$), plus a continuation method that gradually reduces λ, we get the fast iterative shrinkage thresholding algorithm (FISTA).

- Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization. A basic course*. Kluwer.
- Tseng, P. (2008). On Accelerated Proximal Gradient Methods for Convex- Concave Optimization. Technical report, U. Washington.
- Beck, A. and M. Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. on Imaging Sciences 2*(1), 183–202.

EM for Lasso

- For using EM, the key insight is that we can represent the Laplace distribution as a Gaussian scale mixture (GSM)

$$\mathcal{L}ap(w_j|0,1/\gamma) = \frac{\gamma}{2}\exp(-\gamma|w_j|) = \int \mathcal{N}(w_j|0,\tau_j^2)\mathcal{G}a\left(\tau_j^2|1,\frac{\gamma^2}{2}\right)d\tau_j^2$$



Representing lasso using a Gaussian scale mixture prior

- Andrews, D. and C. Mallows (1974). Scale mixtures of Normal distributions. *J. of Royal Stat. Soc. Series B 36*, 99–102.
- West, M. (1987). On scale mixtures of normal distributions. *Biometrika 74*, 646–648.

EM for Lasso

- The corresponding joint distribution has the form

$$p(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{\tau}, \sigma^2)$$

$$= \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \sigma^2\mathbf{I}_N)\mathcal{N}(\boldsymbol{w}|\mathbf{0}, \mathbf{D}_\tau)\mathcal{IG}(\sigma^2|a_\sigma, b_\sigma)\left[\prod_j \mathcal{G}a\left(\tau_j^2|1, \frac{\gamma^2}{2}\right)\right]$$

where $\mathbf{D}_\tau = \text{diag}(\tau_j^2)$.

- We have assumed that $\mathbf{X}$ is standardized and $\boldsymbol{y}$ is centered.
- Expanding out, we get

$$p(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{\tau}, \sigma^2|\mathbf{X})$$

$$\propto (\sigma^2)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\|\boldsymbol{y}\right.$$

Bayesian Lasso

- In the E-step of EM, we infer τ_j^2 and σ^2 and in the M-step, we estimate $\boldsymbol{w}$.

- The latent variable perspective brings several advantages:
  - It provides an easy way to derive an algorithm to find ℓ_1 -regularized parameter estimates for a variety of other models, such as robust linear regression or probit regression.
  - It suggests trying other priors on the variances besides $\mathcal{G}a\left(\tau_j^2|1,\frac{\gamma^2}{2}\right)$.
  - It makes it clear how we can compute the full posterior rather than just the MAP estimate. This technique is known as the Bayesian lasso.

- Figueiredo, M. (2003). Adaptive sparseness for supervised learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence 25*(9), 1150–1159.
- Griffin, J. and P. Brown (2007). Bayesian adaptive lassos with nonconvex penalization. Technical report, U. Kent.
- Caron, F. and A. Doucet (2008). Sparse Bayesian nonparametric regression. In *Intl. Conf. on Machine Learning*.
- Ding, Y. and R. Harrison (2010). A sparse multinomial probit model for classification. *Pattern Analysis and Applications*, 1–9.

Bayesian Lasso

- The complete data penalized log likelihood is as follows (dropping terms that do not depend on $\boldsymbol{w}$)

$$\ell_c(\boldsymbol{w}) = -\frac{1}{2\sigma^2}\|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 - \frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Lambda}\boldsymbol{w} + \text{const}$$

  where $\boldsymbol{\Lambda} = \text{diag}\left(\frac{1}{\tau_j^2}\right)$ is the precision matrix for $\boldsymbol{w}$.

- The key is to compute $\mathbb{E}\left[\frac{1}{\tau_j^2}|w_j\right]$. This can be derived directly.
- Alternatively, we can derive the full posterior, which is given by the following

$$p\left(1/\tau_j^2|\boldsymbol{w},\mathcal{D}\right) = \mathit{InverseGaussian}\left(\sqrt{\frac{\gamma^2}{w_j^2}},\gamma^2\right)$$

- Hence, $\mathbb{E}\left[(1/\tau_j^2)|w_j\right] = \gamma/|w_j|$

- Park, T. and G. Casella (2008). The Bayesian Lasso. *J. of the Am. Stat. Assoc. 103*(482), 681–686.
- Hans, C. (2009). Bayesian Lasso regression. *Biometrika 96*(4), 835– 845.

EM for Lasso

- Let, $\overline{\Lambda} = \text{diag}(\mathbb{E}[1/\tau_1^2], \ldots, \mathbb{E}[1/\tau_D^2])$ denote the result of this E step.
- To infer σ^2, it is easy to show that the posterior is

$$p(\sigma^2|\mathcal{D}, \boldsymbol{w}) = \mathcal{IG}\left(a_\sigma + \frac{N}{2}, b_\sigma + \frac{1}{2}(\boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{w}})^T(\boldsymbol{y} - \mathbf{X}\widehat{\boldsymbol{w}})\right) = \mathcal{IG}(a_{\mathrm{N}}, b_{\mathrm{N}})$$

- Hence,

$$\mathbb{E}\left[\frac{1}{\sigma^2}\right] = \frac{a_N}{b_N} \triangleq \overline{\omega}$$

- The M step consists of computing

$$\widehat{\boldsymbol{w}} = \arg\max_{\boldsymbol{w}}\left(-\frac{1}{2}\overline{\omega}\|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 - \frac{1}{2}\boldsymbol{w}^T\boldsymbol{\Lambda}\mathbf{w}\right)$$

- This is just MAP estimation under a Gaussian prior

$$\widehat{\boldsymbol{w}} = (\sigma^2\overline{\boldsymbol{\Lambda}} + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}\boldsymbol{y}$$

EM for Lasso

- However, since we expect many $w_j = 0$, we will have many $\tau_j^2 = 0$. This makes inverting $\overline{\boldsymbol{\Lambda}}$ numerically unstable.

- Fortunately, we can use the SVD of $\mathbf{X}$, given by $\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^{\mathbf{T}}$ as follows

$$\widehat{\boldsymbol{w}} = \boldsymbol{\Psi}\mathbf{V}\left(\mathbf{V}^T\boldsymbol{\Psi}\mathbf{V} + \frac{1}{\overline{\omega}}\mathbf{D}^{-2}\right)^{-1}\mathbf{D}^{-1}\boldsymbol{U}^T\boldsymbol{y}$$

where

$$\boldsymbol{\Psi} = \overline{\boldsymbol{\Lambda}}^{-1} = \mathrm{diag}\left(\frac{1}{\mathbb{E}\left[1/\tau_j^2\right]}\right) = \mathrm{diag}\left(\frac{|w_j|}{\gamma}\right)$$

- Since the lasso objective is convex, this method should always find the global optimum
  - Unfortunately, this sometimes does not happen, for numerical reasons.
  - In practice, this situation seems to be rare.

- Hunter, D. and R. Li (2005). Variable selection using MM algorithms. *Annals of Statistics 33*, 1617–1642.

ℓ_1 Regularization Extensions

Group Lasso

- In standard ℓ_1 regularization, we assume that there is a 1:1 correspondence between parameters and the variables, so that $\widehat{w}_j = 0$, we interpret this to mean that variable j is excluded.
- But in more complex models, there may be many parameters associated with a given variable.
- In particular, we may have a vector of weights for each input, $\boldsymbol{w}_j$. For example
  - **Multinomial logistic regression:** Each feature is associated with C different weights, one per class.
  - **Linear regression with categorical inputs:** Each scalar input is one-hot encoded into a vector of length C.
  - **Multi-task learning:** In multi-task learning, we have multiple related prediction problems. For example, we might have C separate regression or binary classification problems.

- Obozinski, G., B. Taskar, and M. I Jordan (2007). Joint covariate selection for grouped classification. Technical report, UC Berkeley.

Group Lasso

- If we use an ℓ_1 regularizer of the form $\|\boldsymbol{w}\| = \sum_j \sum_C |w_{jC}|$, we may end up with with some elements of $\boldsymbol{w}_j$ being zero and some not.

- To prevent this kind of situation, we partition the parameter vector into G groups. We now minimize the following objective

$$J(\boldsymbol{w}) = \text{NLL}(\boldsymbol{w}) + \sum_{g=1}^{G} \lambda_g \|\boldsymbol{w}_g\|_2$$

where

$$\|\boldsymbol{w}_g\|_2 = \sqrt{\sum_{j \in \text{g}} w_j^2}$$

is the 2-norm of the group weight vector.

- Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *J. Royal Statistical Society, Series B 68*(1), 49–67.

Group Lasso

- If the NLL is least squares, this method is called group lasso.
- We often use a larger penalty for larger groups, by setting $\lambda_g = \lambda\sqrt{d_g}$ where d_g is the number of elements in group g.
- For example, if we have groups $\{1,2\}$ and $\{3,4,5\}$, the objective becomes

$$J(\boldsymbol{w}) = \mathrm{NLL}(\boldsymbol{w}) + \lambda\left[\sqrt{2}\sqrt{(w_1^2 + w_2^2)} + \sqrt{3}\sqrt{(w_3^2 + w_4^2 + w_5^2)}\right]$$

- Note that if we had used the square of the 2-norms, the model would become equivalent to ridge regression, since

$$\sum_{g=1}^{G} \|\boldsymbol{w}_g\|_2^2 = \sum_{g=1}^{G}\sum_{j\in \mathrm{g}} w_j^2 = \|w\|_2^2$$

Group Lasso

- By using the square root, we are penalizing the radius of a ball containing the group's weight vector:
  - the only way for the radius to be small is if all elements are small
  - Thus the square root results in group sparsity
- A variant of this technique replaces the 2-norm with the infinity-norm (Turlach et al. 2005; Zhao et al. 2005)

$$\|\boldsymbol{w}_g\|_\infty = \max_{j \in \mathrm{g}} |w_j|$$

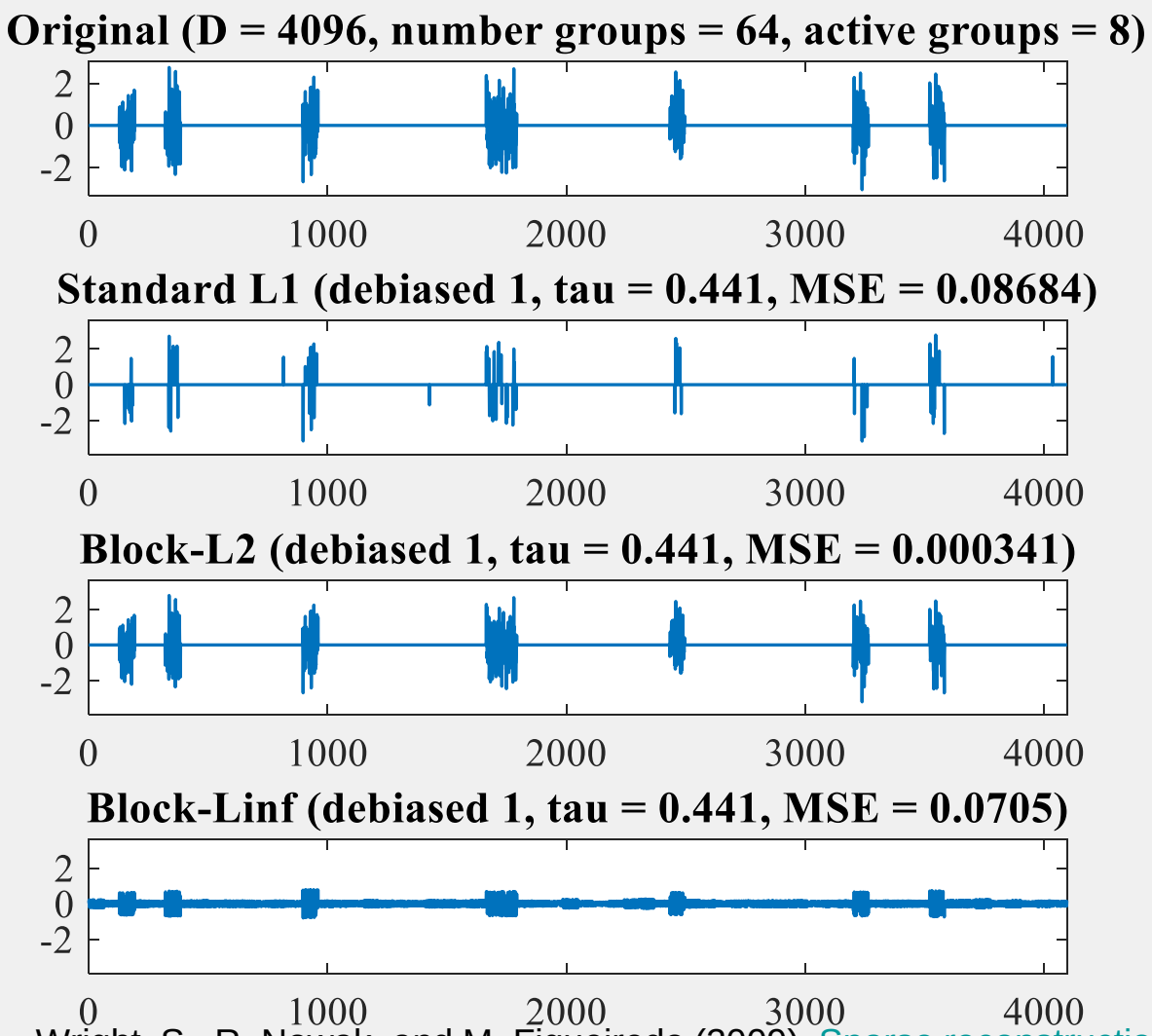
- It is clear that this will also result in group sparsity.

- Turlach, B., W. Venables, and S. Wright (2005). Simultaneous variable selection. *Technometrics 47* (3), 349–363.
- Zhao, P., G. Rocha, and B. Yu (2005). Grouped and Hierarchical Model Selection through Composite Absolute Penalties. Technical report, UC Berkeley.

Group Lasso

- In the following example, we have a true signal $\boldsymbol{w}$ of size $D = 2^{12} = 4096$, divided into 64 groups each of size 64.
- We randomly choose 8 groups of $\boldsymbol{w}$ and assign them non-zero values.
- In the first example, the values are drawn from a $\mathcal{N}(0,1)$. In the second example, the values are all set to 1. We then pick a random design matrix $\boldsymbol{X}$ of size $N \times D$, where $N = 2^{10} = 1024$. Finally, we generate $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{\epsilon}$, $\epsilon \sim \mathcal{N}(0, 10^{-4}\boldsymbol{I}_N)$. Given this data, we estimate the support of $\boldsymbol{w}$ using ℓ_1 or group ℓ_1, and then estimate the non-zero values using least squares.
- We see that group lasso does a much better job than vanilla lasso, since it respects the known group structure. We also see that the ℓ_∞ norm has a tendency to make all the elements within a block to have similar magnitude. This is appropriate in the second example, but not the first.
- The value of λ was the same in all examples and chosen by hand.

Group Lasso

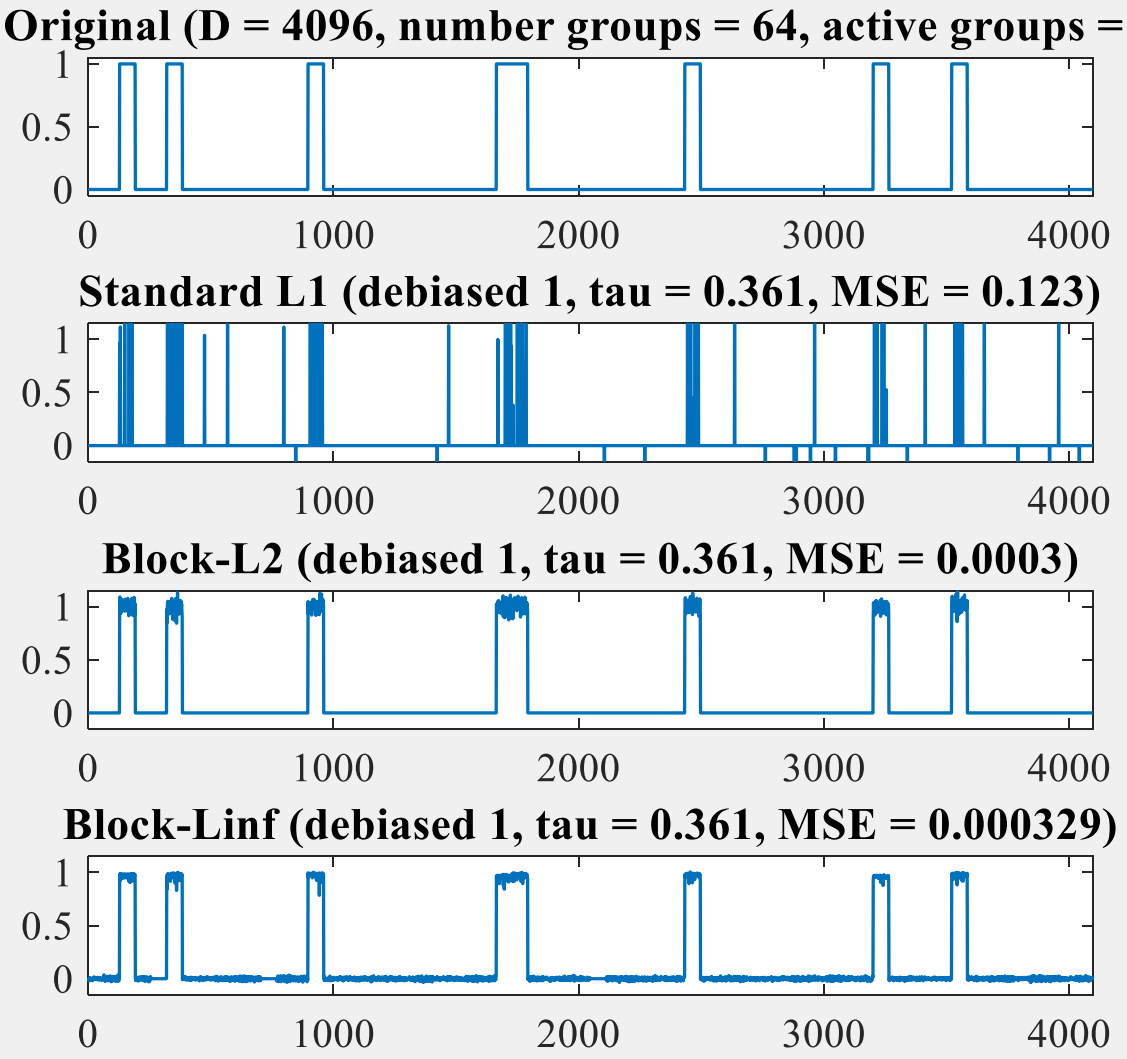


- Illustration of group lasso where the original signal is piecewise Gaussian.
- (a) original signal.
- (b) Vanilla lasso estimate.
- (c) Group lasso estimate using a ℓ_2 norm on the blocks.
- (d) Group lasso estimate using an ℓ_∞norm on the blocks.

- Wright, S., R. Nowak, and M. Figueiredo (2009). Sparse reconstruction by separable approximation. *IEEE Trans. on Signal Processing 57* (7), 2479–2493.

groupLassoDemo.m from PMTK3

Group Lasso



groupLassoDemo.m
from PMTK3

- Illustration of group lasso where the original signal is piecewise constant
- (a) original signal.
- (b) Vanilla lasso estimate.
- (c) Group lasso estimate using a ℓ_2 norm on the blocks.
- (d) Group lasso estimate using an ℓ_∞ norm on the blocks.

GSM Interpretation of Group Lasso

- Group lasso is equivalent to MAP estimation using the following prior

$$p(\boldsymbol{w}|\gamma,\sigma^2) \propto \exp\left(-\frac{\gamma}{\sigma}\sum_{g=1}^{G}\|\boldsymbol{w}_g\|_2\right)$$

- Now one can show that this prior can be written as a Gaussian Scale Mixture (GSM), as follows

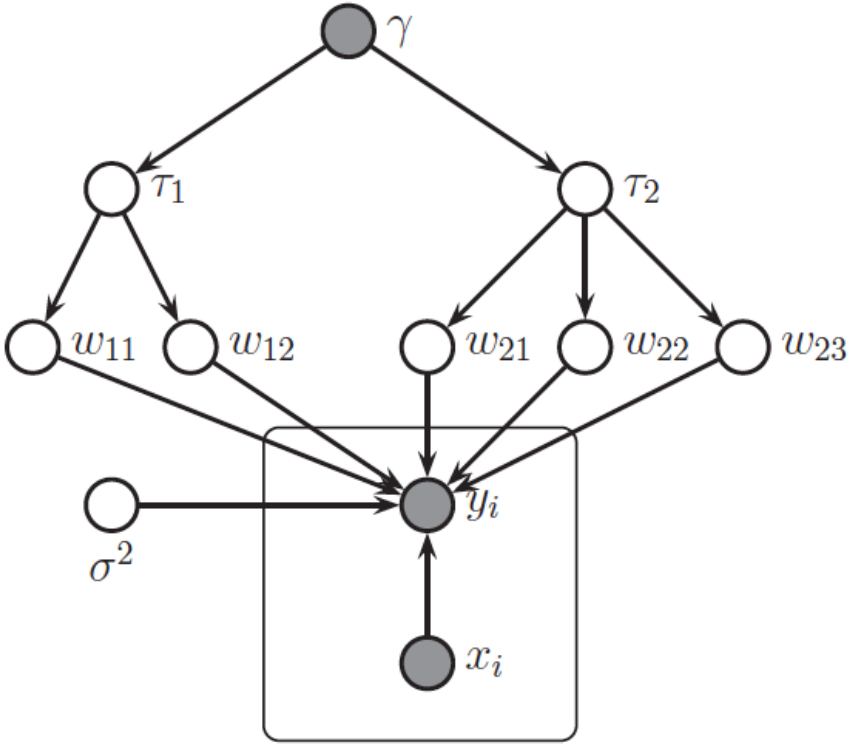
$$w_g|\sigma^2,\tau_g^2 \sim \mathcal{N}\left(0,\sigma^2\tau_g^2\mathbf{I}_{d_g}\right)$$
$$\tau_g^2|\sigma^2 \sim \mathcal{G}a\left(\frac{d_g+1}{2},\frac{\gamma}{2}\right)$$

where d_g is the size of group g.

- So we see that there is one variance term per group, each of which comes from a Gamma prior,
  - shape parameter depends on the group size, and rate parameter is controlled by γ .

GSM Interpretation of Group Lasso

- This picture also makes it clearer why there should be a grouping effect
  - Suppose $w_{1,1}$ is small; then τ_1^2 will be estimated to be small, which will force $w_{1,2}$ to be small
  - Conversely, suppose $w_{1,1}$ is large; then τ_1^2 will be estimated to be large, which will force $w_{1,2}$ to be large



Graphical model for group lasso with 2 groups, the first has size $G_1 = 2$, the second has size $G_2 = 3$.

Algorithms for Group Lasso

- We discuss two algorithms for group lasso.
- The first approach is based on proximal gradient descent.
- Since the regularizer is separable, $R(\boldsymbol{w}) = \sum_g \left\| \boldsymbol{w}_g \right\|_p$, the proximal operator decomposes into G separate operators of the form:

$$\text{prox}_R(\boldsymbol{b}) = \arg\min_{z \in \mathbb{R}^{D_g}} \|\boldsymbol{z} - \boldsymbol{b}\|_2^2 + \lambda \|\boldsymbol{z}\|_p$$

where $\boldsymbol{b} = \boldsymbol{\theta}_{kg} - t_k \boldsymbol{g}_{kg}$.

- If $p = 2$, one can show that this can be implemented as follows

$$\text{prox}_R(\boldsymbol{b}) = \boldsymbol{b} - \text{prox}_{\lambda C}(\boldsymbol{b})$$

where $C = \{\boldsymbol{z}: \|\boldsymbol{z}\|_2 \leq 1\}$ is the ℓ_2 ball.

- Using $\text{proj}_C(\boldsymbol{\theta}) = \begin{cases} \theta / \|\theta\|_2, & \|\theta\|_2 > 1 \\ \theta, & \|\theta\|_2 \leq 1 \end{cases}$, if $\|\boldsymbol{b}\|_2 < \lambda$, we have

$$\text{prox}_R(\boldsymbol{b}) = \boldsymbol{b} - \boldsymbol{b} = \boldsymbol{0}$$

- Combettes, P. and V. Wajs (2005). Signal recovery by proximal forwardbackward splitting. *SIAM J. Multiscale Model. Simul. 4*(4), 1168–1200.

Algorithms for Group Lasso

- Otherwise we have

$$\text{prox}_R(\boldsymbol{b}) = \boldsymbol{b} - \lambda \frac{\boldsymbol{b}}{\|\boldsymbol{b}\|_2} = \boldsymbol{b} \frac{\|\boldsymbol{b}\|_2 - \lambda}{\|\boldsymbol{b}\|_2}$$

- We can combine these into a vectorial soft-threshold function as follows (Wright et al. 2009)

$$\text{prox}_R(\boldsymbol{b}) = \boldsymbol{b} \frac{\max(\|\boldsymbol{b}\|_2 - \lambda, 0)}{\max(\|\boldsymbol{b}\|_2 - \lambda, 0) + \lambda}$$

- If $p \to \infty$, we use $C = \{\boldsymbol{z}: \|\boldsymbol{z}\|_1 \leq 1\}$ which is the ℓ_1 ball.
- We can project onto this in $\mathcal{O}(d_g)$ time using an algorithm described in (Duchi et al. 2008).
- Another approach is to modify the EM algorithm.
- The method is almost the same as for vanilla lasso.
- If we define $\tau_j^2 = \tau_{g(j)}^2$, $g(j)$ is the group to which dimension j belongs, we can use the same full conditionals for σ^2 and $\boldsymbol{w}$ as before.

- Wright, S., R. Nowak, and M. Figueiredo (2009). Sparse reconstruction by separable approximation. *IEEE Trans. on Signal Processing 57* (7), 2479–2493.
- Duchi, J., S. Shalev-Shwartz, Y. Singer, and T. Chandra (2008). Efficient projections onto the L1-ball for learning in high dimensions. In *Intl. Conf. on Machine Learning*.

Algorithms for Group Lasso

- The only changes are as follows
  - We must modify the full conditional for the weight precisions, which are estimated based on a shared set of weights

$$\frac{1}{\tau_g^2}|\gamma, \boldsymbol{w}, \sigma^2, \boldsymbol{y}, \mathbf{X} \sim \mathit{InverseGaussian}\left(\sqrt{\frac{\gamma^2\sigma^2}{\|\boldsymbol{w}_g\|_2^2}}, \gamma^2\right)$$

where $\|\boldsymbol{w}_g\|_2^2 = \sum_{j \in \mathrm{g}} w_{jg}^2$.

  - For the E step, we can use

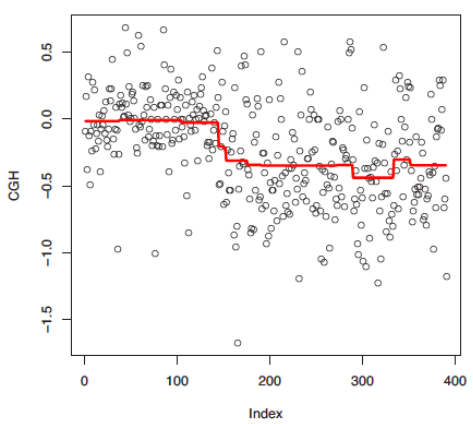
$$\mathbb{E}\left[\frac{1}{\tau_g^2}\right] = \frac{\gamma\sigma}{\|w_g\|_2}$$

  - We must modify the full conditional for the tuning parameter, which is now only estimated based on G values τ_g^2

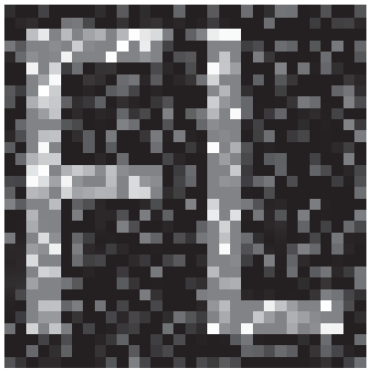
$$p(\gamma^2|\tau) = \mathcal{G}a\left(a_\gamma + \frac{G}{2}, b_\gamma + 0.5\sum_g^G \tau_g^2\right)$$

Fused Lasso

- In some problem settings (e.g., functional data analysis), we want neighboring coefficients to be similar to each other, in addition to being sparse.



(a)



(b)

(c)

(a) Example of the fused lasso. The vertical axis represents array CGH (chromosomal genome hybridization) intensity, and the horizontal axis represents location along a genome.
(b) Noisy image.
(c) Fused lasso estimate using 2d lattice prior.

- Hoefling, H. (2010). A Path Algorithm for the Fused Lasso Signal Approximator. Technical report, Stanford

Fused Lasso

- In some problem settings (e.g., functional data analysis), we want neighboring coefficients to be similar to each other, in addition to being sparse.
- An example is given in Fig. (a) (previous slide), where we want to fit a signal that is mostly "off", but in addition has the property that neighboring locations are typically similar in value.
- We can model this by using a prior of the form

$$p(\boldsymbol{w}|\sigma^2) \propto \exp\left(-\frac{\lambda_1}{\sigma}\sum_{j}^{D}|w_j| - \frac{\lambda_2}{\sigma}\sum_{j}^{D-1}|w_{j+1} - w_j|\right)$$

- This is known as the fused lasso penalty.
- In the context of functional data analysis, we often use $\mathbf{X} = \mathbf{I}$, so there is one coefficient for each location in the signal.

Fused Lasso

- In this case, the overall objective has the form

$$J(\boldsymbol{w}, \lambda_1, \lambda_2) = \sum_{i=1}^{N} (y_i - w_i)^2 + \lambda_1 \sum_{i=1}^{N} |w_i| + \lambda_2 \sum_{i=1}^{N-1} |w_{i+1} - w_i|$$

- It is possible to generalize this idea beyond chains, and to consider other graph structures, using a penalty of the form

$$J(\boldsymbol{w}, \lambda_1, \lambda_2) = \sum_{s \in \mathrm{V}} (y_s - w_s)^2 + \lambda_1 \sum_{s \in \mathrm{V}} |w_s| + \lambda_2 \sum_{(s,t) \in E} |w_s - w_t|$$

- This is called graph-guided fused lasso.
- The graph might come from some prior knowledge, e.g., from a database of known biological pathways.

- Chen, X., S. Kim, Q. Lin, J. G. Carbonell, and E. P. Xing (2010). Graph-Structured Multi-task Regression and an Efficient Optimization Method for General Fused Lasso. Technical report, CMU.

GSM Interpretation of Fused Lasso

- One can show (Kyung et al. 2010) that the fused lasso model is equivalent to the following hierarchical model

$$\boldsymbol{w}|\sigma^2, \boldsymbol{\tau}, \boldsymbol{\omega} \sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \boldsymbol{\Sigma}(\boldsymbol{\tau}, \boldsymbol{\omega})\right)$$

$$\tau_j^2|\gamma_1 \sim \mathcal{E}xp\mathcal{o}n\left(\frac{\gamma_1^2}{2}\right), \qquad j = 1{:}\,D$$

$$\omega_j^2|\gamma_2 \sim \mathcal{E}xp\mathcal{o}n\left(\frac{\gamma_2^2}{2}\right), \qquad j = 1{:}\,D-1$$

where

$$\boldsymbol{\Sigma} = \boldsymbol{\Omega}^{-1}$$

and $\boldsymbol{\Omega}$ is a tridiagonal precision matrix with

$$\text{Main diagonal} = \left\{\frac{1}{\tau_j^2} + \frac{1}{\omega_{j-1}^2} + \frac{1}{\omega_j^2}\right\}$$

$$\text{Off diagonal} = \left\{-\frac{1}{\omega_j^2}\right\}$$

where we have defined $\omega_0^{-2} = \omega_D^{-2} = 0$.

- Kyung, M., J. Gill, M. Ghosh, and G. Casella (2010). Penalized Regression, Standard Errors and Bayesian Lassos. *Bayesian Analysis* 5(2), 369–412.

GSM Interpretation of Fused Lasso

- In the case of graph-guided lasso, the structure of the graph is reflected in the zero pattern of the Gaussian precision matrix.
- It is possible to generalize the EM algorithm to fit the fused lasso model, by exploiting the Markov structure of the Gaussian prior for efficiency.
- Direct solvers (which don't use the latent variable trick) can also be derived (see e.g., (Hoefling 2010)).
- However, this model is more expensive to fit than the other variants we have considered.

- Hoefling, H. (2010). A Path Algorithm for the Fused Lasso Signal Approximator. Technical report, Stanford

Elastic Net (Ridge & Lasso Combined)

- Although lasso has proved to be effective as a variable selection technique, it has several problems:
  - If there is a group of variables that are highly correlated (e.g., genes that are in the same pathway), then the lasso tends to select only one of them, chosen rather arbitrarily. It is usually better to select all the relevant variables in a group. If we know the grouping structure, we can use group lasso, but often we don't know the grouping structure.
  - If $D > N$, lasso can select at most N variables before it saturates.
  - If $N > D$, but the variables are correlated, it has been empirically observed that the prediction performance of ridge is better than that of lasso.
- Elastic net, which is an hybrid between ridge and lasso addresses some of these issues.
- It is apparently called the "elastic net" because it is "like a stretchable fishing net that retains 'all the big fish"'.

- Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *J. of Royal Stat. Soc. Series B 67* (2), 301–320.

Elastic Net (Ridge & Lasso Combined)

- The vanilla version of the elastic net defines the following objective function

$$J(\boldsymbol{w}, \lambda_1, \lambda_2) = \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|^2 + \lambda_2 \|\boldsymbol{w}\|_2^2 + \lambda_1 \|\boldsymbol{w}\|_1$$

- Notice that this penalty function is strictly convex (assuming $\lambda_2 > 0$) so there is a global unique minima, even if $\mathbf{X}$ is not full-rank.
- It can be shown (Zou and Hastie 2005) that any strictly convex penalty on $\boldsymbol{w}$ will exhibit a grouping effect, which means that the regression coefficients of highly correlated variables tend to be equal (up to a change of sign if they are negatively correlated).
- For example, if $\mathbf{X}_{:,k} = \mathbf{X}_{:,j}$, one can show that there estimates are equal, $\hat{w}_k = \hat{w}_j$.
- By contrast, with lasso we may have $\hat{w}_k = 0$ and $\hat{w}_j \neq 0$ or vice versa.

- Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *J. of Royal Stat. Soc. Series B 67* (2), 301–320.

LARS-Elastic Net Algorithm

- It is simple to show that the elastic net problem can be reduced to a lasso problem on modified data. In particular, define

$$\tilde{\mathbf{X}} = c\begin{pmatrix}\mathbf{X}\\ \sqrt{\lambda_2}\mathbf{I}_D\end{pmatrix}, \qquad \tilde{\boldsymbol{y}} = \begin{pmatrix}\boldsymbol{y}\\ \mathbf{0}_{D\times 1}\end{pmatrix}$$

where

$$c = (1+\lambda_2)^{-0.5}$$

- Then we solve

$$\tilde{\boldsymbol{w}} = \arg\min_{\tilde{\boldsymbol{w}}} \left\|\tilde{\boldsymbol{y}} - \tilde{\mathbf{X}}\tilde{\boldsymbol{w}}\right\|^2 + c\lambda_1\|\boldsymbol{w}\|_1$$

and set

$$\boldsymbol{w} = c\tilde{\boldsymbol{w}}$$

- We can use LARS to solve this subproblem; this is known as the LARS-EN algorithm.

- If we stop the algorithm after m variables have been included, the cost is $\mathcal{O}(m^3 + Dm^2)$.

LARS-Elastic Net Algorithm

- If we stop the algorithm after m variables have been included, the cost is $\mathcal{O}(m^3 + Dm^2)$.
- Note that we can use $m = D$ if we wish, since $\tilde{\mathbf{X}}$ has rank D.
- This is in contrast to lasso, which cannot select more than N variables.
- When using LARS-EN (or other ℓ_1 solvers), one typically uses cross-validation to select λ_1 and λ_2.
- Unfortunately it turns out that the "vanilla" elastic net does not produce functions that predict very accurately, unless it is very close to either pure ridge or pure lasso.
- Intuitively the reason is that it performs shrinkage twice: once during ℓ_1 penalty an once during ℓ_2 penalty.
- The solution is simple: undo the ℓ_2 shrinkage by scaling up the estimates from the vanilla version.

LARS-Elastic Net Algorithm

$$\widehat{\boldsymbol{w}} = \sqrt{(1+\lambda_2)}\widetilde{\boldsymbol{w}}$$

- We will call this a corrected estimate
- One can show that the corrected estimates are given by

$$\widehat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} \boldsymbol{w}^T \left(\frac{\mathbf{X}^T\mathbf{X} + \lambda_2 \mathbf{I}}{1+\lambda_2} \right) \boldsymbol{w} - 2\boldsymbol{y}^T\mathbf{X}\boldsymbol{w} + \lambda_1 \|\boldsymbol{w}\|_1$$

- Now

$$\frac{\mathbf{X}^T\mathbf{X} + \lambda_2 \mathbf{I}}{1+\lambda_2} = (1-\rho)\widehat{\boldsymbol{\Sigma}} + \rho\mathbf{I}$$

where

$$\rho = \frac{\lambda_2}{1+\lambda_2}$$

- So the the elastic net is like lasso but where we use a version of $\widehat{\boldsymbol{\Sigma}}$ that is shrunk towards $\mathbf{I}$.

GSM Interpretation of Elastic Net

- The implicit prior being used by the elastic net obviously has the form

$$p(\boldsymbol{w}|\sigma^2) \propto \exp\left(-\frac{\gamma_1}{\sigma}\sum_{j=1}^{D}|w_j| - \frac{\gamma_2}{2\sigma^2}\sum_{j=1}^{D}w_j^2\right)$$

- This is just a product of Gaussian and Laplace distributions
- It can be written as a hierarchical prior (Kyung et al. 2010; Chen et al. 2011)

$$w_j|\sigma^2, \tau_j^2 \sim \mathcal{N}\left(0, \sigma^2\left(\tau_j^{-2} + \gamma_2\right)^{-1}\right)$$
$$\tau_j^2|\gamma_1 \sim \mathcal{E}xp\mathcal{o}n(\gamma_1^2/2)$$

- Clearly, if $\gamma_2 = 0$, this reduces to regular lasso.
- Can perform MAP estimation using EM, or Bayesian inference using MCMC (Kyung et al. 2010) or variational Bayes (Chen et al. 2011).

- Kyung, M., J. Gill, M. Ghosh, and G. Casella (2010). Penalized Regression, Standard Errors and Bayesian Lassos. *Bayesian Analysis 5*(2), 369–412.
- Chen, M., D. Carlson, A. Zaas, C. Woods, G. Ginsburg, A. Hero, J. Lucas, and L. Carin (2011, March). The Bayesian Elastic Net: Classifying Multi-Task Gene-Expression Data. *IEEE Trans. Biomed. Eng. 58*(3), 468–79.

Non-Convex Regularizers

Non-Convex Regularizers

- Although the Laplace prior results in a convex optimization problem, from a statistical point of view this prior is not ideal
  - First, it does not put enough probability mass near 0, so it doesn't sufficiently suppress noise.
  - Second, it does not put enough probability mass on large values, so it causes shrinkage of relevant coefficients, corresponding to "signal".
- Both problems can be solved by going to more flexible kinds of priors which have a larger spike at 0 and heavier tails.
- Even though we cannot find the global optimum anymore, these non-convex methods often outperform ℓ_1 regularization, both in terms of predictive accuracy and in detecting relevant variables.

- Fan, J. and R. Z. Li (2001). Variable selection via non-concave penalized likelihood and its oracle properties. *J. of the Am. Stat. Assoc. 96*(456), 1348–1360.
- Seeger, M. (2008). Bayesian Inference and Optimal Design in the Sparse Linear Model. *J. of Machine Learning Research 9*, 759–813

Bridge Regression

- A natural generalization of ℓ_1 regularization, known as bridge regression (Frank and Friedman 1993), has the form

$$\hat{\boldsymbol{w}} = \text{NLL}(\boldsymbol{w}) + \lambda \sum_j |w_j|^b$$

for $b \geq 0$.

- This corresponds to MAP estimation using an exponential power distribution

$$\mathcal{ExpPower}(x|\mu, a, b) \triangleq \frac{b}{2a\Gamma\left(1+\frac{1}{b}\right)} \exp\left(-\frac{|x-\mu|^b}{a}\right)$$

  - If $b = 2$, we get Gaussian distribution (with $a = \sigma\sqrt{2}$) corresponding to ridge regression
  - If $b = 1$, we get the Laplace distribution, corresponding to lasso.
  - If $b = 0$, we get ℓ_0 regression, which is equivalent to best subset selection.

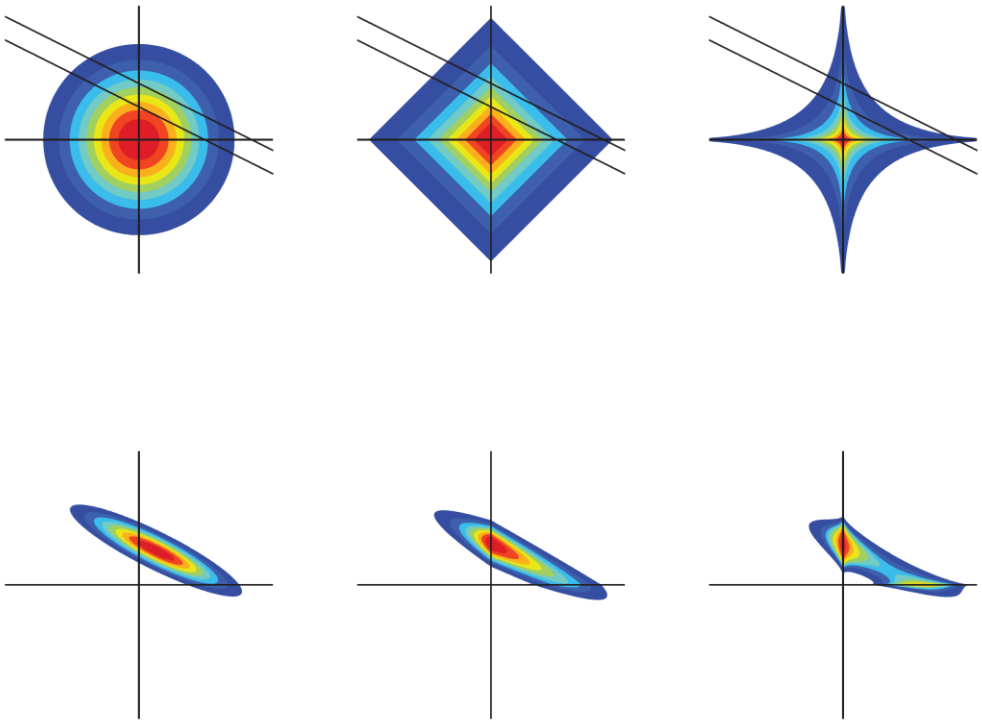
- Frank, I. and J. Friedman (1993). A statistical view of some chemometrics regression tools. *Technometrics 35*(2), 109–135.

Bridge Regression

- Unfortunately, the objective is not convex for $b < 1$ and not sparsity promoting for $b > 1$.
- So, ℓ_1 norm is the tightest convex approximation to the ℓ_0 norm.

sparsePostPlot.m
from PMTK3

- Top: plot of log prior for three different distributions with unit variance: Gaussian, Laplace and exponential power.
- Bottom: plot of log posterior after observing a single observation, corresponding to a single linear constraint. The precision of this observation is shown by the diagonal lines in the top figure.



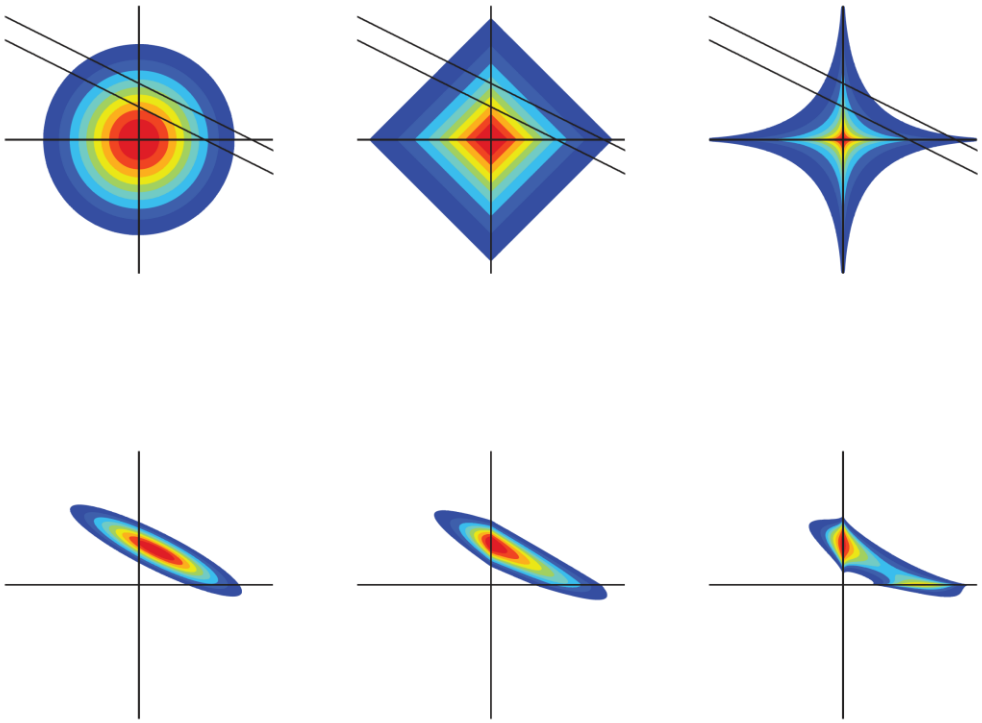
- Seeger, M. and H. Nickish (2008). Compressed sensing and Bayesian experimental design. In *Intl. Conf. on Machine Learning*.

Bridge Regression

- Unfortunately, the objective is not convex for $b < 1$ and not sparsity promoting for $b > 1$.
- So, ℓ_1 norm is the tightest convex approximation to the ℓ_0 norm.

sparsePostPlot.m from PMTK3

- In the case of the Gaussian prior, the posterior is unimodal and symmetric.
- In the case of the Laplace prior, the posterior is unimodal and asymmetric (skewed).
- In the case of the exponential prior, the posterior is bimodal.



- Seeger, M. and H. Nickish (2008). Compressed sensing and Bayesian experimental design. In *Intl. Conf. on Machine Learning*.

Hierarchical Adaptive Lasso

- Recall that one of the principal problems with lasso is that it results in biased estimates
  - This is because Lasso uses a large value of λ to "squash" the irrelevant parameters, but this then over-penalizes the relevant parameters.
- It would be better if we could associate a different penalty parameter with each parameter.
- Of course, it is completely infeasible to tune D parameters by cross validation, but this poses no problem to the Bayesian.
- We simply make each τ_j^2 have its own private tuning parameter, γ_j, which are now treated as random variables coming from the conjugate prior $\gamma_j \sim \mathcal{IG}(a, b)$.

- Lee, A., F. Caron, A. Doucet, and C. Holmes (2010). A hierarchical bayesian framework for constructing sparsity-inducing priors. Technical report, U. Oxford.

Hierarchical Adaptive Lasso

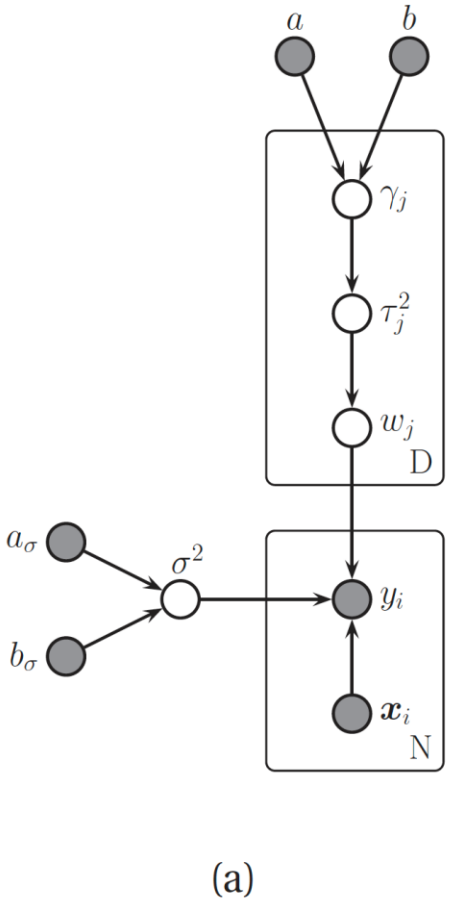
- The full model is as follows

$$\gamma_j \sim \mathcal{IG}(a, b)$$
$$\tau_j^2 | \gamma_j \sim \mathcal{G}a\left(1, \gamma_j^2/2\right)$$
$$w_j | \tau_j^2 \sim \mathcal{N}\left(0, \tau_j^2\right)$$

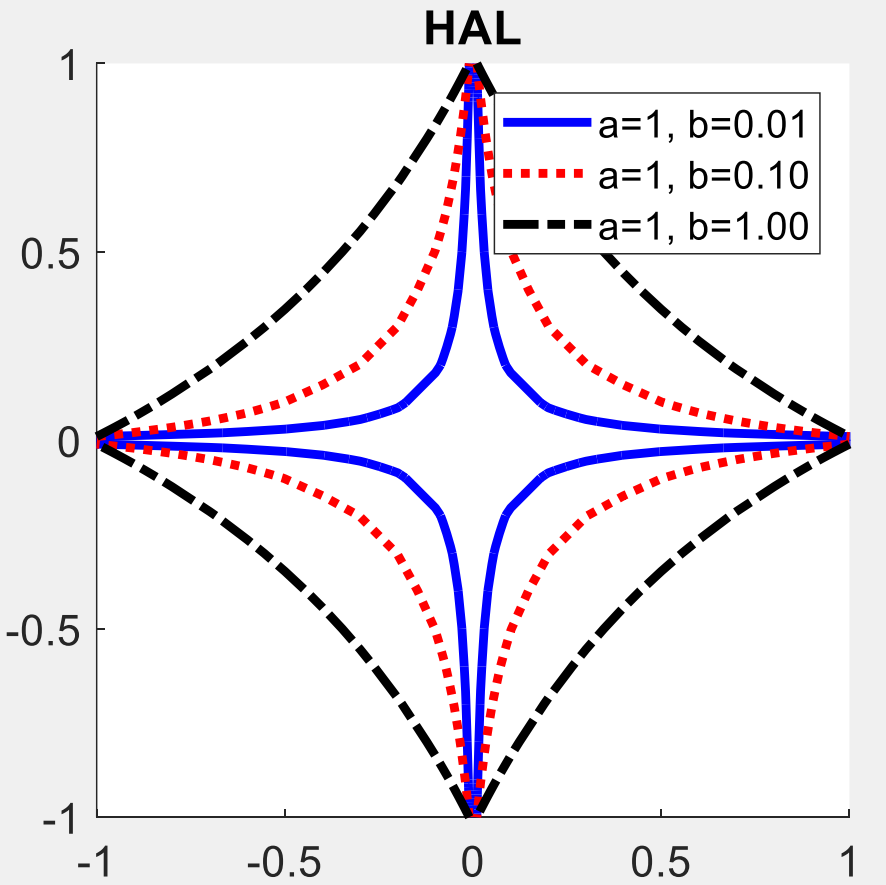
- We can integrate out τ_j^2 which induces a $\mathcal{L}ap\left(w_j | 0, 1/\gamma_j\right)$ on w_j as before.
- As a result $p\left(w_j\right)$ is now a scaled mixture of Laplacians.
- It turns out that we can fit this model (i.e., compute a local posterior mode) using EM.
- The resulting estimate $\widehat{\boldsymbol{w}}_{HAL}$, often works much better than the estimate returned by lasso $\widehat{\boldsymbol{w}}_{L1}$.
  - it is more likely to contain zeros in the right places (model selection consistency) and more likely to result in good predictions (prediction consistency) (Lee et al. 2010).

- Lee, A., F. Caron, A. Doucet, and C. Holmes (2010). A hierarchical bayesian framework for constructing sparsity-inducing priors. Technical report, U. Oxford.

Hierarchical Adaptive Lasso (HAL)



(a)



normalGammaPenaltyPlotDemo.m
from PMTK3

(a) DGM for hierarchical adaptive lasso. (b) Contours of Hierarchical adaptive Laplace.

EM for HAL

- Since the inverse Gamma is conjugate to the Laplace, we find that the E step for γ_j is given by

$$p(\gamma_j|w_j) = \mathcal{IG}(a+1, b+|w_j|)$$

- The E step for σ^2 is the same as for vanilla lasso.

- The prior for $\boldsymbol{w}$ has the following form

$$p(\boldsymbol{w}|\gamma) = \prod_j \frac{1}{2\gamma_j} \exp(-|w_j|/\gamma_j)$$

- Hence the M step must optimize

$$\hat{\boldsymbol{w}}^{(t+1)} = \arg\max_{\boldsymbol{w}} \log \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \sigma^2) - \sum_j |w_j| \mathbb{E}[1/\gamma_j]$$

- The expectation is given by

$$\mathbb{E}[1/\gamma_j] = \frac{a+1}{b+|w_j^{(t)}|} \triangleq s_j^{(t)}$$

EM for HAL

- Thus the M step becomes a weighted lasso problem

$$\widehat{w}^{(t+1)} = \arg\min_{\boldsymbol{w}} \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + \sum_j s_j^{(t)} |w_j|$$

- This is easily solved using standard methods (e.g., LARS).
- Note that if the coefficient was estimated to be large in the previous iteration (so $w_j^{(t)}$ is large), then the scaling factor $s_j^{(t)}$ will be small.
  - so large coefficients are not penalized heavily.
- Conversely, small coefficients do get penalized heavily.
- This is the way that the algorithm adapts the penalization strength of each coefficient.
- The result is an estimate that is often much sparser than returned by lasso, but also less biased.

EM for HAL

- Note that if we set $a = b = 0$, and we only perform 1 iteration of EM, we get a method that is closely related to the adaptive lasso of (Zou 2006; Zou and Li 2008).

- This EM algorithm is also closely related to some iteratively reweighted ℓ_1 methods proposed in the signal processing community (Chartrand and Yin 2008; Candes et al. 2008).

- We can get a better understanding of HAL by integrating out γ_j to get the following marginal distribution,

$$p(w_j|a,b) = \frac{a}{2b}\left(\frac{|w_j|}{b} + 1\right)^{-(a+1)}$$

- Zou, H. (2006). The adaptive Lasso and its oracle properties. *J. of the Am. Stat. Assoc.*, 1418–1429.
- Zou, H. and R. Li (2008). One step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics 36*(4), 1509–1533.
- Chartrand, R. and W. Yin (2008). Iteratively reweighted algorithms for compressive sensing. In *Intl. Conf. on Acoustics, Speech and Signal Proc.*
- Candes, E. and M. Wakin (2008,March). An introduction to compressive sampling. *IEEE Signal Processing Magazine 21*.

EM for HAL

$$p(w_j|a,b) = \frac{a}{2b}\left(\frac{|w_j|}{b}+1\right)^{-(a+1)}$$

- This is an instance of the generalized $\mathcal{T}$ distribution (McDonald and Newey 1988) (in (Cevher 2009; Armagan et al. 2011), this is called the double Pareto distribution) defined as

$$\mathcal{GT}(w|\mu,,c,q) \triangleq \frac{q}{2ca^{1/q}B(1/q,a)}\left(1+\frac{|w-\mu|^q}{ac^q}\right)^{-(a+1/q)}$$

where c is the scale parameter (which controls the degree of sparsity), and a is related to the degrees of freedom.

- Zou, H. (2006). The adaptive Lasso and its oracle properties. *J. of the Am. Stat. Assoc.*, 1418–1429.
- Zou, H. and R. Li (2008). One step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics 36*(4), 1509–1533.
- Chartrand, R. and W. Yin (2008). Iteratively reweighted algorithms for compressive sensing. In *Intl. Conf. on Acoustics, Speech and Signal Proc.*
- Candes, E. and M. Wakin (2008,March). An introduction to compressive sampling. *IEEE Signal Processing Magazine 21*.

EM for HAL

$$\mathcal{GT}(w|\mu,,c,q) \triangleq \frac{q}{2ca^{1/q}B(1/q,a)}\left(1+\frac{|w-\mu|^q}{ac^q}\right)^{-(a+1/q)}$$

- When $q=2$ and $c=\sqrt{2}$, we recover the standard $\mathcal{T}$ distribution.
- When $a \to \infty$, we recover the exponential power distribution.
- When $q=1$ and $a \to \infty$ we get the Laplace distribution.
- In the context of the current model, we see that $p(w_j|a,b) = \mathcal{GT}(w_j|0,a,b/a,1)$.
- The resulting penalty term has the form

$$\pi_{\boldsymbol{\lambda}}(w_j) \triangleq -\log p(w_j) = (a+1)\log\left(1+\frac{|w_j|}{b}\right) + \text{const}$$

- McDonald, J. and W. Newey (1988). Partially Adaptive Estimation of Regression Models via the Generalized t Distribution. *Econometric Theory 4*(3), 428–445.
- Cevher, V. (2009). Learning with compressible priors. In *NIPS*.
- Armagan, A., D. Dunson, and J. Lee (2011). Generalized double pareto shrinkage. Technical report, Duke.

EM for HAL

- $\boldsymbol{\lambda} = (a, b)$ are the tuning parameters.
- We understand the behavior of this penalty function by applying it to the problem of linear regression with an orthogonal design matrix.
- In this case, the objective becomes

$$J(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + \sum_{j=1}^{D} \pi_{\boldsymbol{\lambda}}(|w_j|)$$

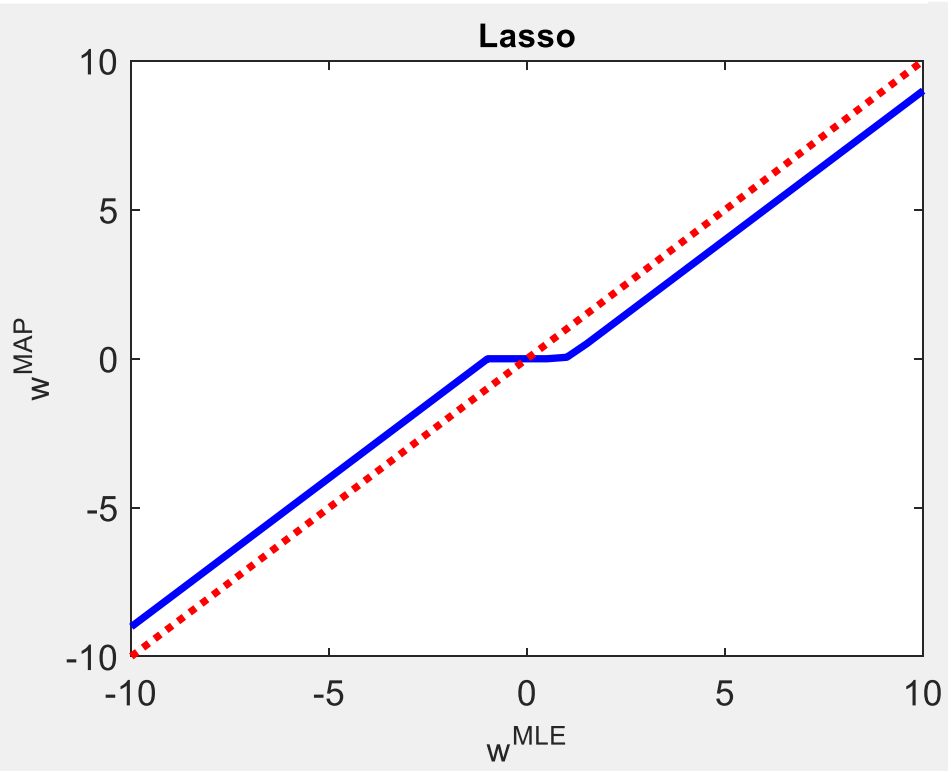
$$= \frac{1}{2}\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 + \frac{1}{2}\sum_{j=1}^{D}(\hat{w}_j^{mle} - w_j)^2 + \sum_{j=1}^{D} \pi_{\boldsymbol{\lambda}}(|w_j|)$$

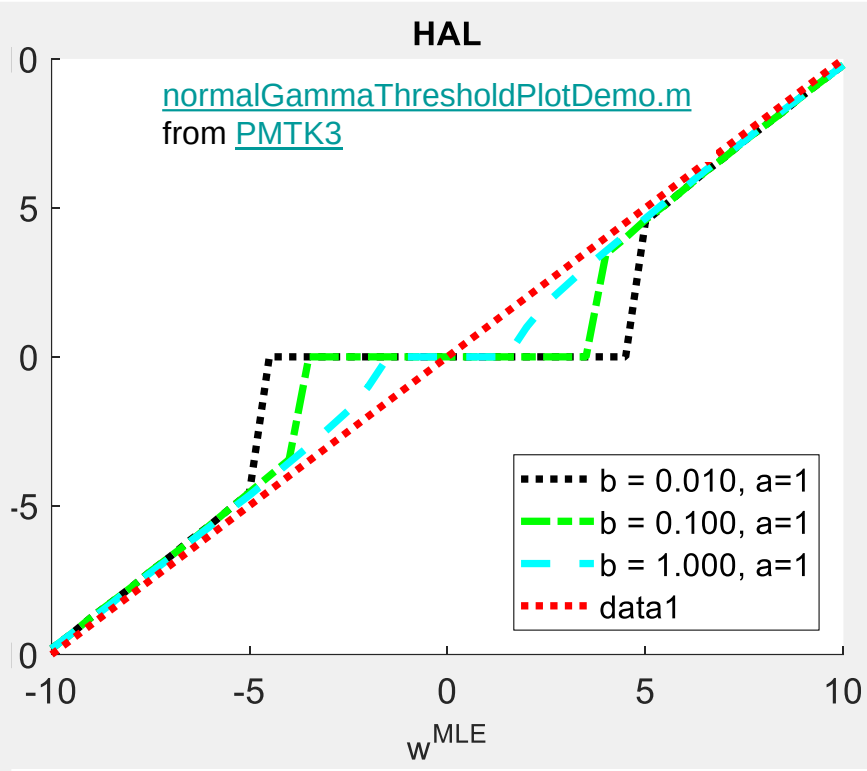
where $\hat{\boldsymbol{w}}^{mle} = \mathbf{X}^T\boldsymbol{y}$ is the MLE and $\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{w}}^{mle}$

- Thus we can compute the MAP estimate one dimension at a time by solving the following 1d optimization problem

$$\hat{w}_j = \arg\min_{w_j} 0.5(\hat{w}_j^{mle} - w_j)^2 + \pi_{\boldsymbol{\lambda}}(w_j)$$

EM for HAL





Thresholding behavior of two penalty functions (negative log priors). (a) Laplace. (b) Hierarchical adaptive Laplace

- Lee, A., F. Caron, A. Doucet, and C. Holmes (2010). A hierarchical bayesian framework for constructing sparsity-inducing priors. Technical report, U. Oxford.

Other Hierarchical Priors

$p(\tau_j^2)$	$p(\gamma_j)$	$p(w_j)$	Ref
$\text{Ga}(1, \frac{\gamma^2}{2})$	Fixed	$\text{Lap}(0, 1/\gamma)$	(Andrews and Mallows 1974; West 1987)
$\text{Ga}(1, \frac{\gamma^2}{2})$	$\text{IG}(a, b)$	$\text{GT}(0, a, b/a, 1)$	(Lee et al. 2010, 2011; Cevher 2009; Armagan et al. 2011)
$\text{Ga}(1, \frac{\gamma^2}{2})$	$\text{Ga}(a, b)$	$\text{NEG}(a, b)$	(Griffin and Brown 2007, 2010; Chen et al. 2011)
$\text{Ga}(\delta, \frac{\gamma^2}{2})$	Fixed	$\text{NG}(\delta, \gamma)$	(Griffin and Brown 2007, 2010)
$\text{Ga}(\tau_j^2 \vert 0, 0)$	-	$\text{NJ}(w_j)$	(Figueiredo 2003)
$\text{IG}(\frac{\delta}{2}, \frac{\delta\gamma^2}{2})$	Fixed	$\mathcal{T}(0, \delta, \gamma)$	(Andrews and Mallows 1974; West 1987)
$C^+(0, \gamma)$	$C^+(0, b)$	$\text{horseshoe}(b)$	(Carvahlo et al. 2010)

- Andrews, D. and C. Mallows (1974). Scale mixtures of Normal distributions. *J. of Royal Stat. Soc. Series B 36*, 99–102.
- West, M. (1987). On scale mixtures of normal distributions. *Biometrika 74*, 646–648.
- Lee, A., F. Caron, A. Doucet, and C. Holmes (2010). A hierarchical bayesian framework for constructing sparsity-inducing priors. Technical report, U. Oxford.
- Lee, A., F. Caron, A. Doucet, and C. Holmes (2011). Bayesian Sparsity-Path-Analysis of Genetic Association Signal using Generalized t Prior. Technical report, U. Oxford.
- Cevher, V. (2009). Learning with compressible priors. In *NIPS*.
- Armagan, A., D. Dunson, and J. Lee (2011). Generalized double pareto shrinkage. Technical report, Duke.
- Griffin, J. and P. Brown (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis 5*(1), 171– 188.
- Griffin, J. and P. Brown (2007). Bayesian adaptive lassos with nonconvex penalization. Technical report, U. Kent.
- Chen, M., D. Carlson, A. Zaas, C. Woods, G. Ginsburg, A. Hero, J. Lucas, and L. Carin (2011, March). The Bayesian Elastic Net: Classifying Multi-Task Gene-Expression Data. *IEEE Trans. Biomed. Eng. 58*(3), 468–79.
- Figueiredo, M. (2003). Adaptive sparseness for supervised learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence 25*(9), 1150–1159.
- Carvahlo, C., N. Polson, and J. Scott (2010). The horseshoe estimator for sparse signals. *Biometrika 97* (2), 465.

Other Hierarchical Priors

$p(\tau_j^2)$	$p(\gamma_j)$	$p(w_j)$	Ref
$\text{Ga}(1, \frac{\gamma^2}{2})$	Fixed	$\text{Lap}(0, 1/\gamma)$	(Andrews and Mallows 1974; West 1987)
$\text{Ga}(1, \frac{\gamma^2}{2})$	$\text{IG}(a, b)$	$\text{GT}(0, a, b/a, 1)$	(Lee et al. 2010, 2011; Cevher 2009; Armagan et al. 2011)
$\text{Ga}(1, \frac{\gamma^2}{2})$	$\text{Ga}(a, b)$	$\text{NEG}(a, b)$	(Griffin and Brown 2007, 2010; Chen et al. 2011)
$\text{Ga}(\delta, \frac{\gamma^2}{2})$	Fixed	$\text{NG}(\delta, \gamma)$	(Griffin and Brown 2007, 2010)
$\text{Ga}(\tau_j^2 \vert 0, 0)$	-	$\text{NJ}(w_j)$	(Figueiredo 2003)
$\text{IG}(\frac{\delta}{2}, \frac{\delta\gamma^2}{2})$	Fixed	$\mathcal{T}(0, \delta, \gamma)$	(Andrews and Mallows 1974; West 1987)
$C^+(0, \gamma)$	$C^+(0, b)$	$\text{horseshoe}(b)$	(Carvahlo et al. 2010)

Some scale mixtures of Gaussians.

- ✓ $\mathcal{C}^+$ = half-rectified Cauchy;
- ✓ $\mathcal{G}a$ = Gamma (shape and rate parameterization);
- ✓ $\mathcal{GT}$ = generalized t;
- ✓ $\mathcal{IG}$ = inverse Gamma;
- ✓ $\mathcal{NEG}$ = Normal-Exponential-Gamma;
- ✓ $\mathcal{NG}$ = Normal-Gamma;
- ✓ $\mathcal{NJ}$ = Normal-Jeffreys.
- ✓ The $horseshoe\ distribution$ is the distribution induced on w_j by the prior described in (Carvahlo et al. 2010).

Automatic Relevance Determination

Automatic Relevance Determination (ARD)

- We discuss an alternative approach based on type II ML estimation (empirical Bayes), whereby we integrate out $\boldsymbol{w}$ and maximize the marginal likelihood wrt $\boldsymbol{\tau}$.

- This EB procedure can be implemented via EM or via a reweighted ℓ_1 scheme.

- Having estimated the variances, we plug them in to compute the posterior mean of the weights $\mathbb{E}[\boldsymbol{w}|\hat{\boldsymbol{\tau}}, \mathcal{D}]$.

- Rather surprisingly (in view of the Gaussian prior), the result is an (approximately) sparse estimate.

- T. Fletcher, Relevance Vector Machines Explained, October 2010.

Automatic Relevance Determination (ARD)

- In the context of neural networks, this method is called called automatic relevance determination or ARD (MacKay 1995b; Neal 1996)
- In the context of linear models this method is called sparse Bayesian learning or SBL (Tipping 2001)
- Combining ARD/SBL with basis function expansion in a linear model gives rise to a technique called the relevance vector machine (RVM).

- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. of Machine Learning Research 1*, 211–244.
- MacKay, D. (1995b). Probable networks and plausible predictions — a review of practical Bayesian methods for supervised neural networks. *Network*.
- Neal, R. (1996). *Bayesian learning for neural networks*. Springer.

ARD for Linear Regression

- We will explain the procedure in the context of linear regression.
- ARD for GLMs requires the use of the Laplace (or some other) approximation.
- It is conventional, when discussing ARD / SBL, to denote the weight precisions by $\alpha_j = 1/\tau_j^2$, and the measurement precision by $\beta = \frac{1}{\sigma^2}$.
- In particular, we will assume the following model

$$p(y|\boldsymbol{x}, \boldsymbol{w}, \beta) = \mathcal{N}(y|\boldsymbol{w}^T\boldsymbol{x}, 1/\beta)$$

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \mathbf{A}^{-1})$$

where $\mathbf{A} = \text{diag}(\boldsymbol{\alpha})$.

ARD for Linear Regression

- The marginal likelihood can be computed analytically as follows

$$p(\boldsymbol{y}|\mathbf{X},\boldsymbol{\alpha},\beta) = \int \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w},\beta\mathbf{I}_N)\mathcal{N}(\boldsymbol{w}|\mathbf{0},\mathbf{A})d\boldsymbol{w} = \mathcal{N}(\boldsymbol{y}|\mathbf{0},\beta\mathbf{I}_N + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T)$$
$$= (2\pi)^{-N/2}|\mathbf{C}_{\boldsymbol{\alpha}}|^{-0.5}\exp(-0.5\boldsymbol{y}^T\mathbf{C}_{\boldsymbol{\alpha}}^{-1}\boldsymbol{y})$$

where

$$\mathbf{C}_{\boldsymbol{\alpha}} \triangleq \beta^{-1}\mathbf{I}_N + \mathbf{X}\mathbf{A}^{-1}\mathbf{X}^T$$

- Compare this to the marginal likelihood in the spike and slab model
  - modulo the $\beta = 1/\sigma^2$ factor missing from the second term,
  - the equations are the same, except we have replaced the binary $\gamma_j = \{0,1\}$ with continuous $\alpha_j \in \mathbb{R}^+$.

ARD for Linear Regression

- In log form, the objective becomes

$$\ell(\boldsymbol{\alpha}, \beta) \triangleq -\frac{1}{2}\log p\big((\boldsymbol{y}|\mathbf{X}, \boldsymbol{\alpha}, \beta)\big) = \log|\mathbf{C}_{\boldsymbol{\alpha}}| + \boldsymbol{y}^T \mathbf{C}_{\boldsymbol{\alpha}}^{-1} \boldsymbol{y}$$

- To regularize the problem, we may put a conjugate prior on each precision, $\alpha_j \sim \mathcal{G}a(a, b)$ and $\beta \sim \mathcal{G}a(c, d)$.

- The modified objective becomes

$$\ell(\boldsymbol{\alpha}, \beta) \triangleq -\frac{1}{2}\log p\big((\boldsymbol{y}|\mathbf{X}, \boldsymbol{\alpha}, \beta)\big) + \sum_j \log \mathcal{G}a\big(\alpha_j|a, b\big) + \log \mathcal{G}a(\beta|c, d)$$

$$= \log|\mathbf{C}_{\boldsymbol{\alpha}}| + \boldsymbol{y}^T \mathbf{C}_{\boldsymbol{\alpha}}^{-1} \boldsymbol{y} + \sum_j \big(a \log \alpha_j - b\alpha_j\big) + c \log \beta - d\beta$$

- This is useful when performing Bayesian inference for $\boldsymbol{\alpha}$ and β.

  - However, when performing (type II) point estimation, we will use the improper prior $a = b = c = d = 0$ which results in maximal sparsity.

- Bishop, C. and M. Tipping (2000). Variational relevance vector machines. In *UAI*.
- Buntine, W. and A. Weigend (1991). Bayesian backpropagation. *Complex Systems 5*, 603–643.

ARD for Linear Regression

- Once we have estimated $\boldsymbol{\alpha}$ and β, we can compute the posterior over the parameters using

$$p\left(\boldsymbol{w}|\mathcal{D}, \hat{\boldsymbol{\alpha}}, \hat{\beta}\right) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$
$$\boldsymbol{\Sigma}^{-1} = \hat{\beta}\mathbf{X}^T\mathbf{X} + \mathbf{A}$$
$$\boldsymbol{\mu} = \hat{\beta}\boldsymbol{\Sigma}\mathbf{X}^T\boldsymbol{y}$$

- The fact that we compute a posterior over $\boldsymbol{w}$, while simultaneously encouraging sparsity, is why the method is called "sparse Bayesian learning".

- Nevertheless, since there are many ways to be sparse and Bayesian, we will use the "ARD" term instead, even in the linear model context.

- In addition, SBL is only "being Bayesian" about the values of the coefficients, rather than reflecting uncertainty about the set of relevant variables, which is typically of more interest.

Sparsity

- If $\hat{\alpha}_j \approx 0$, we find $\hat{w}_j \approx \hat{w}_j^{mle}$, since the Gaussian prior shrinking w_j towards 0 has zero precision.
- However, if we find that $\hat{\alpha}_j \to \infty$ then the prior is very confident that $w_j = 0$, and hence that feature j is "irrelevant".
- Hence the posterior mean will have $\hat{w}_j \approx 0$.
- Thus irrelevant features automatically have their weights "turned off" or "pruned out.
- We now give an intuitive argument, based on (Tipping 2001), about why ML-II should encourage $\alpha_j \to \infty$ for irrelevant features.
- Consider a 1d linear regression with 2 training examples, so $\mathbf{X} = \mathbf{x} = (x_1, x_2)$ and $\boldsymbol{y} = (y_1, y_2)$.
- We can plot $\mathbf{x}$ and $\boldsymbol{y}$ as vectors in the plane.

- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. of Machine Learning Research 1*, 211–244.

ARD and Sparsity

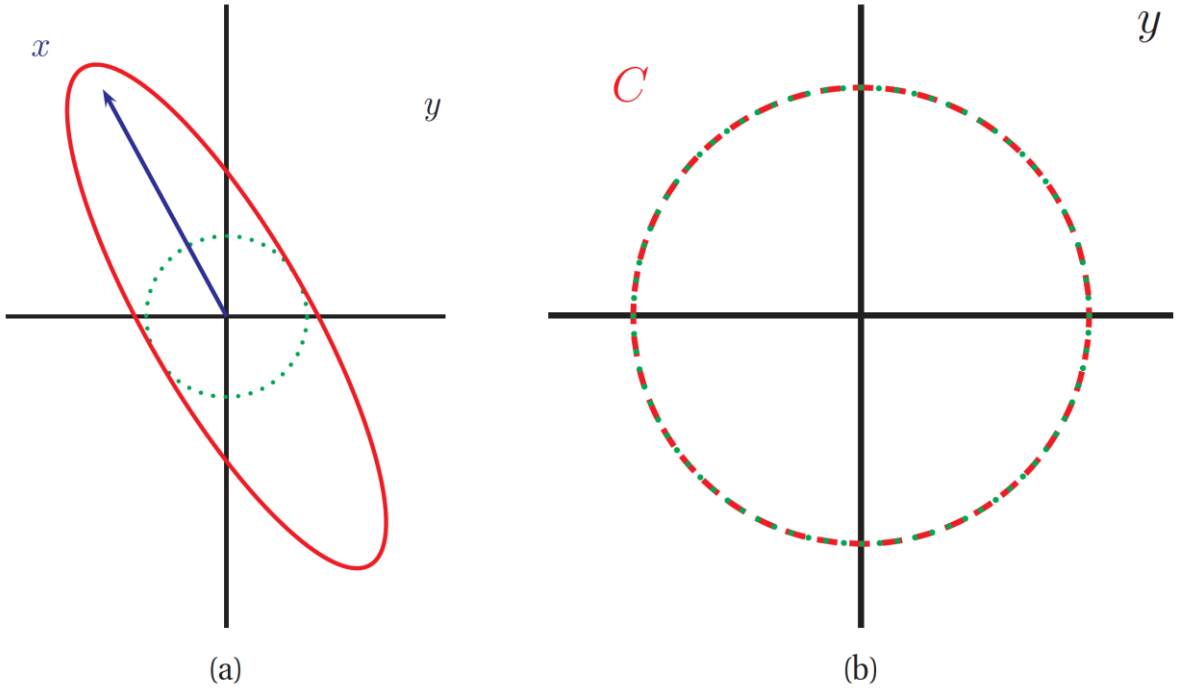


Illustration of why ARD results in sparsity. The vector of inputs $\boldsymbol{x}$ does not point towards the vector of outputs $\boldsymbol{y}$, so the feature should be removed.

(a) For finite α, the probability density is spread in directions away from y.

(b) When $\alpha = \infty$, the probability density at y is maximized.

- Tipping, M. (2001). Sparse Bayesian learning and the relevance vector machine. *J. of Machine Learning Research 1*, 211–244 (see Fig. 8).

Sparsity

- Suppose the feature is irrelevant for predicting the response, so $\boldsymbol{x}$ points in a nearly orthogonal direction to $\boldsymbol{y}$.
- Let us see what happens to the marginal likelihood as we change α.
- The marginal likelihood is given by $p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \mathbf{C})$ where

$$\mathbf{C} = \frac{1}{\beta}\mathbf{I} + \frac{1}{\alpha}\boldsymbol{x}\boldsymbol{x}^T$$

- If α is finite, the posterior will be elongated along the direction of $\boldsymbol{x}$. However, if $\alpha \to \infty$, we find $\mathbf{C} = \frac{1}{\beta}\mathbf{I}$.
  - So, $\mathbf{C}$ is spherical.
- If $|\mathbf{C}|$ is held constant, the latter assigns higher probability density to the observed response vector $\boldsymbol{y}$, so this is the preferred solution.

Sparsity

- In other words, the marginal likelihood "punishes" solutions where α_j is small but $\mathbf{X}_{:,j}$ is irrelevant, since these waste probability mass.

- It is more parsimonious (from the point of view of Bayesian Occam's razor) to eliminate redundant dimensions.

Connection to MAP Estimation

- ARD seems quite different from the MAP estimation methods we considered earlier.
- In particular, in ARD, we are not integrating out α and optimizing $\boldsymbol{w}$, but vice versa.
- Because the parameters w_j become correlated in the posterior (due to explaining away), when we estimate α_j, we are borrowing information from all the features, not just feature j.
- Consequently, the effective prior $p(\boldsymbol{w}|\widehat{\boldsymbol{\alpha}})$ is non-factorial.
- Furthermore it depends on the data $\mathcal{D}$ (and σ^2).
- However, in (Wipf and Nagarajan 2007), it was shown that ARD can be viewed as the following MAP estimation problem

$$\widehat{\boldsymbol{w}}^{ARD} = \arg\min_{\boldsymbol{w}} \beta \|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|_2^2 + g_{ARD}(\boldsymbol{w})$$

$$g_{ARD}(\boldsymbol{w}) \triangleq \min_{\alpha \geq 0} \sum_j \alpha_j w_j^2 + \log|\mathbf{C}_{\boldsymbol{\alpha}}|$$

- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted $\ell-1$ and $\ell-2$ Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).

Connection to MAP Estimation

- Furthermore, (Wipf and Nagarajan 2007; Wipf et al. 2010) prove that MAP estimation with non-factorial priors is strictly better than MAP estimation with any possible factorial prior.

- The non-factorial objective always has fewer local minima than factorial objectives, while still satisfying the property that the global optimum of the non-factorial objective corresponds to the global optimum of the ℓ_0 objective

  - ℓ_1 regularization, which has no local minima, does not enjoy this property.

- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted ℓ−1 and ℓ−2 Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).
- Wipf, D. and S. Nagarajan (2007). A new view of automatic relevancy determination. In *NIPS*.

EM for ARD

- The expected complete data log likelihood is given by

$$Q(\boldsymbol{\alpha}, \beta) = \mathbb{E}[\log \mathcal{N}(\boldsymbol{y}|\mathbf{X}\boldsymbol{w}, \sigma^2\mathbf{I}) + \log \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \mathbf{A}^{-1})]$$
$$= \frac{1}{2}\mathbb{E}\left[N \log \beta - \beta\|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|^2 + \sum_j \log \alpha_j - \mathrm{tr}(\mathbf{A}\boldsymbol{w}\boldsymbol{w}^T)\right] + \text{const}$$
$$= \frac{1}{2}N \log \beta - \frac{\beta}{2}\left(\|\boldsymbol{y} - \mathbf{X}\boldsymbol{w}\|^2 + tr(\mathbf{X}^T\mathbf{X}\boldsymbol{\Sigma})\right) + \frac{1}{2}\sum_j \log \alpha_j - \frac{1}{2}\mathrm{tr}[\mathbf{A}(\boldsymbol{\mu}\boldsymbol{\mu}^T + \boldsymbol{\Sigma})]$$
$$+ \text{const}$$

$\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are computed in the E-step using $\boldsymbol{\Sigma}^{-1} = \hat{\beta}\mathbf{X}^T\mathbf{X} + \mathbf{A}, \boldsymbol{\mu} = \hat{\beta}\boldsymbol{\Sigma}\mathbf{X}^T\boldsymbol{y}$.

- Suppose we put a $\mathcal{G}a(a, b)$ prior on α_j and a $\mathcal{G}a(c, d)$ prior on β.

- The penalized objective becomes

$$Q'^{(\boldsymbol{\alpha},\beta)} = Q(\boldsymbol{\alpha}, \beta) + \sum_j \left(a \log \alpha_j - b\alpha_j\right) + c \log \beta - d\beta$$

EM for ARD

- Setting $\frac{dQ'}{d\alpha_j} = 0$, we get the following M step

$$\alpha_j = \frac{1 + 2a}{\mathbb{E}\left[w_j^2\right] + 2b} = \frac{1 + 2a}{m_j^2 + \Sigma_{jj} + 2b}$$

- If $\alpha_j = \alpha$ and $a = b = 0$, the update becomes

$$\alpha = \frac{D}{\mathbb{E}[\boldsymbol{w}^T \boldsymbol{w}]} = \frac{D}{\boldsymbol{\mu}^T \boldsymbol{\mu} + \mathrm{tr}(\boldsymbol{\Sigma})}$$

- The update for β is given as

$$\beta_{new}^{-1} = \frac{\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\mu}\|^2 + \beta^{-1} \sum_j \left(1 - \alpha_j \Sigma_{jj}\right) + 2d}{N + 2c}$$

Fixed-point Algorithm for ARD

- A faster and more direct approach is to directly optimize the objective

$$\ell(\boldsymbol{\alpha}, \beta) \triangleq \log|\mathbf{C}_{\boldsymbol{\alpha}}| + \boldsymbol{y}^T \mathbf{C}_{\boldsymbol{\alpha}}^{-1} \boldsymbol{y} + \sum_j \left(a \log \alpha_j - b\alpha_j\right) + c \log \beta - d\beta$$

- One can show that the equations $\frac{d\ell}{d\alpha_j} = 0$ and $\frac{d\ell}{d\beta} = 0$ lead to the following fixed point updates

$$\alpha_j \leftarrow \frac{\gamma_j + 2a}{m_j^2 + 2b}$$

$$\beta^{-1} \leftarrow \frac{\|\boldsymbol{y} - \mathbf{X}\boldsymbol{\mu}\|^2 + 2d}{N - \sum_j \gamma_j + 2c}$$

$$\gamma_j \triangleq 1 - \alpha_j \Sigma_{jj}$$

- The quantity γ_j is a measure of how well-determined w_j is by the data (MacKay 1992).

- Hence, $\gamma = \sum_j \gamma_j$ is the effective degrees of freedom of the model.

- MacKay, D. (1999). Comparision of approximate methods for handling hyperparameters. *Neural Computation 11*(5), 1035–1068.
- Wipf, D. and S. Nagarajan (2007). A new view of automatic relevancy determination. In *NIPS*

Fixed-point Algorithm for ARD

- Since α and β both depend on μ and Σ, we need to re-estimate these equations until convergence.

- At convergence, the results are formally identical to those obtained by EM, but

  - ✓ since the objective is non-convex, the results can depend on the initial values.

Iteratively Reweighted ℓ_1 Algorithm

- Another approach to solving the ARD problem is based on the view that it is a MAP estimation problem.
- Although the log prior $g(\boldsymbol{w})$ is rather complex in form, it can be shown to be a non-decreasing, concave function of $|w_j|$.
- This means that it can be solved by an iteratively reweighted ℓ_1 problem of the form

$$\boldsymbol{w}^{t+1} = \arg\min_{\boldsymbol{w}} \text{NLL}(\boldsymbol{w}) + \sum_j \lambda_j^{(t)} |w_j|$$

- In (Wipf and Nagarajan 2010), the following procedure for setting the penalty terms is suggested
  - We initialize with $\lambda_j^{(0)} = 1$ and then at iteration $t+1$ compute $\lambda_j^{(t+1)}$ by iterating the following equation a few times

$$\lambda_j \leftarrow \left[\mathbf{X}_{:,j} \left(\sigma^2 \mathbf{I} + \left(\mathbf{X} \text{diag}\left(\frac{1}{\lambda_j}\right) \text{diag}\left(\left| w_j^{(t+1)} \right| \right) \right)^{-1} \mathbf{X}^T \right)^{-1} \mathbf{X}_{:,j} \right]^{\frac{1}{2}}$$

- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted ℓ−1 and ℓ−2 Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).

Iteratively Reweighted ℓ_1 Algorithm

- We see that the new penalty λ_j depends on all the old weights.
- To understand this difference, consider the noiseless case where $\sigma^2 = 0$, and assume $D \gg N$.
- In this case, there are $\binom{D}{N}$ solutions which perfectly reconstruct the data $\mathbf{X}\boldsymbol{w} = \boldsymbol{y}$, and which have sparsity $\|\boldsymbol{w}\|_0 = N$.
- These are called basic feasible solutions or BFS.
- What we want are solutions that satisfy $\mathbf{X}\boldsymbol{w} = \boldsymbol{y}$, but which are much sparser than this
  - We do not want to increase the penalty on a weight just because it is small (as in adaptive lasso), since that will just reinforce our current local optimum.
  - Instead, we want to increase the penalty on a weight if it is small and if we have $\left\|w^{(t+1)}\right\| < N$.

- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted ℓ−1 and ℓ−2 Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).

Iteratively Reweighted ℓ_1 Algorithm

- The covariance term $\left(\mathbf{X}\text{diag}\left(\frac{1}{\lambda_j}\right)\text{diag}\left(\left|w_j^{(t+1)}\right|\right)\right)^{-1}$ has this effect
  - if $\boldsymbol{w}$ is a BFS, this matrix will be full rank so the penalty will not increase much
  - but if $\boldsymbol{w}$ is sparser than N, the matrix will not be full rank, so the penalties associated with zero-valued coefficients will increase, thus reinforcing this solution (Wipf and Nagarajan 2010).

- Wipf, D. and S. Nagarajan (2010, April). Iterative Reweighted ℓ−1 and ℓ−2 Methods for Finding Sparse Solutions. *J. of Selected Topics in Signal Processing (Special Issue on Compressive Sensing) 4*(2).
- Wipf, D. and S. Nagarajan (2007). A new view of automatic relevancy determination. In *NIPS*.

ARD for Logistic Regression

- Now consider binary logistic regression $p(y|\boldsymbol{x}, \boldsymbol{w}) = \mathcal{B}er\left(y|\text{sigm}(\boldsymbol{w}^T\boldsymbol{x})\right)$ with same Gaussian prior $p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mathbf{0}, \mathbf{A}^{-1})$.
- We can no longer use EM to estimate $\boldsymbol{\alpha}$, since the Gaussian prior is not conjugate to the logistic likelihood, so the E step cannot be done exactly.
- One approach is to use a variational approximation to the E step.
- A simpler approach is to use a Laplace approximation in the E-step.
- We can then use this approximation inside the same EM procedure as before, except we no longer need to update β.
  - Note, however, that this is not guaranteed to converge
- An alternative is to use the iteratively reweighted ℓ_1 algorithm discussed before.
- In this case, no approximation is needed.

Sparse Priors for Unsupervised Learning

- We now focus on using sparse priors for unsupervised learning.
- One popular option for unsupervised learning is ICA.
- ICA is similar to PCA except that it uses a non-Gaussian prior for the latent factors $\boldsymbol{z}_i$.
- If we make the non-Gaussian prior be sparsity promoting, such as a Laplace distribution, we will be approximating each observed vector $\boldsymbol{x}_i$ as a sparse combination of basis vectors (columns of $\boldsymbol{W}$)
  - note that the sparsity pattern changes from data case to data case
- If we relax the constraint that $\mathbf{W}$ is orthogonal, we get a method called sparse coding.
- In this context, we call the factor loading matrix $\mathbf{W}$ a dictionary, each column is referred to as an atom.

Sparse Coding

Sparse Coding

- In view of the sparse representation, it is common for $L > D$, in which case we call the representation overcomplete.
- In sparse coding, the dictionary can be fixed or learned.
- If it is fixed, it is common to use a wavelet or DCT basis since many natural signals can be well approximated by a small number of such basis functions.
- However, it is also possible to learn the dictionary, by maximizing the likelihood

$$\log p(\mathcal{D}|\mathbf{W}) = \sum_{i=1}^{N} \int_{\boldsymbol{z}_i} \mathcal{N}(\boldsymbol{x}_i|\mathbf{W}\boldsymbol{z}_i, \sigma^2\mathbf{I}) p(\boldsymbol{z}_i) d\boldsymbol{z}_i$$

- Do not confuse sparse coding with sparse PCA
  - Sparse PCA puts a sparsity promoting prior on the regression weights W
  - Sparse coding puts a sparsity promoting prior on the latent factors $\boldsymbol{z}_i$.

- Witten, D., R. Tibshirani, and T. Hastie (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics 10*(3), 515–534.
- Journee, M., Y. Nesterov, P. Richtarik, and R. Sepulchre (2010). Generalized power method for sparse principal components analysis. *J. of Machine Learning Research 11*, 517– 553.

Sparse Coding

- Of course, the two techniques can be combined; we call the result sparse matrix factorization

Method	$p(\mathbf{z}_i)$	$p(\mathbf{W})$	$\mathbf{W}$ orthogonal
PCA	Gauss	-	yes
FA	Gauss	-	no
ICA	Non-Gauss	-	yes
Sparse coding	Laplace	-	no
Sparse PCA	Gauss	Laplace	maybe
Sparse MF	Laplace	Laplace	no

Summary of various latent factor models. A dash "-" in the $p(\boldsymbol{W})$ column means we are performing ML parameter estimation rather than MAP parameter estimation

Learning a Sparse Coding Dictionary

- We make the following approximation

$$\log p(\mathcal{D}|\mathbf{W}) \approx \sum_{i=1}^{N} \max_{\boldsymbol{z}_i} [\log \mathcal{N}(\boldsymbol{x}_i|\mathbf{W}\boldsymbol{z}_i, \sigma^2\mathbf{I}) + \log p(\boldsymbol{z}_i)]$$

- If $p(\boldsymbol{z}_i)$ is Laplace, we can rewrite the NLL as

$$\text{NLL}(\mathbf{W}, \mathbf{Z}) = \sum_{i=1}^{N} \frac{1}{2} \|\boldsymbol{x}_i - \mathbf{W}\boldsymbol{z}_i\|_2^2 + \lambda \|\boldsymbol{z}_i\|_1$$

- To prevent $\mathbf{W}$ from becoming arbitrarily large, it is common to constrain the ℓ_2 norm of its columns to be less than or equal to 1.

$$\mathcal{C} = \left\{\mathbf{W} \in \mathbb{R}^{D \times L} \quad \text{s.t.} \quad \boldsymbol{w}_j^T \boldsymbol{w}_j \leq 1\right\}$$

- Then we want to solve $\min_{\mathbf{W} \in \mathcal{C}, \mathbf{Z} \in \mathbb{R}^{N \times L}} \text{NLL}(\mathbf{W}, \mathbf{Z})$

- For a fixed $\boldsymbol{z}_i$, the optimization over $\mathbf{W}$ is a simple least squares problem.

- Mumford, D. (1994). Neuronal architectures for pattern-theoretic problems. In C. Koch and J. Davis (Eds.), *Large Scale Neuronal Theories of the Brain*. MIT Press.

Learning a Sparse Coding Dictionary

- For a fixed dictionary $\mathbf{W}$, the optimization problem over $\mathbf{Z}$ is identical to the lasso problem, for which many fast algorithms exist.
- This suggests an iterative optimization: alternate between optimizing $\mathbf{W}$ and $\mathbf{Z}$.
- (Mumford 1994) called this kind of approach an analysis-synthesis
  - estimating the basis $\mathbf{W}$ is the analysis phase
  - estimating the coefficients $\mathbf{Z}$ is the synthesis phase
- If this is too slow, other algorithms can be used (Mairal et al. 2010).
- A variety of other models result in an optimization problem that looks similar to learning a sparse coding dictionary
  - E.g., non-negative matrix factorization or NMF (Paatero and Tapper 1994; Lee and Seung 2001) requires solving an objective of the form

$$\min_{\mathbf{W}\in\mathcal{C},\mathbf{Z}\in\mathbb{R}^{L\times N}} \frac{1}{2}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \mathbf{W}\boldsymbol{z}_i\|_2^2, \qquad \text{s.t. } \mathbf{W}\geq 0, \boldsymbol{z}_i \geq 0$$

- Mumford, D. (1994). Neuronal architectures for pattern-theoretic problems. In C. Koch and J. Davis (Eds.), *Large Scale Neuronal Theories of the Brain*. MIT Press.
- Mairal, J., F. Bach, J. Ponce, and G. Sapiro (2010). Online learning for matrix factorization and sparse coding. *J. of Machine Learning Research 11*, 19–60.
- Paatero, P. and U. Tapper (1994). Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values. *Environmetrics 5*, 111–126.
- Lee, D. and S. Seung (2001). Algorithms for non-negative matrix factorization. In *NIPS*.

Learning a Sparse Coding Dictionary

$$\min_{\mathbf{W}\in\mathcal{C},\mathbf{Z}\in\mathbb{R}^{L\times N}} \frac{1}{2}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \mathbf{W}\boldsymbol{z}_i\|_2^2, \qquad \text{s.t.}\, \mathbf{W} \geq 0, \boldsymbol{z}_i \geq 0$$

- Note that this has no hyper-parameters to tune.
- The intuition behind this constraint is that the learned dictionary may be more interpretable if it is a positive sum of positive "parts", rather than a sparse sum of atoms that may be positive or negative.
- Of course, we can combine NMF with a sparsity promoting prior on the latent factors. This is called non-negative sparse coding.
- Alternatively, we can drop the positivity constraint, but impose a sparsity constraint on both the factors $\boldsymbol{z}_i$ and the dictionary $\mathbf{W}$.
  - We call this sparse matrix factorization.

- Hoyer, P. (2004). Non-negative matrix factorizaton with sparseness constraints. *J. of Machine Learning Research 5*, 1457–1469.

Learning a Sparse Coding Dictionary

- To ensure strict convexity, we can use an elastic net type penalty on the weights

$$\min_{\mathbf{W},\mathbf{Z}} \frac{1}{2}\sum_{i=1}^{N} \|\boldsymbol{x}_i - \mathbf{W}\boldsymbol{z}_i\|_2^2 + \lambda\|\boldsymbol{z}_i\|_1 , \qquad \text{s.t.} \left\|\boldsymbol{w}_j\right\|_2^2 + \gamma\left\|\boldsymbol{w}_j\right\|_1 \leq 1$$

- There are several related objectives one can write down
  - For example, we can replace the lasso NLL with group lasso or fused lasso (Witten et al. 2009)
- We can also use other sparsity-promoting priors besides the Laplace.
- For example, (Zhou et al. 2009) propose a model in which the latent factors $\boldsymbol{z}_i$ are made sparse using the binary mask model.
  - Each bit of the mask can be generated from a Bernoulli distribution with parameter π, which can be drawn from beta distribution.

- Zhou, H., D. Karakos, S. Khudanpur, A. Andreou, and C. Priebe (2009). On Projections of Gaussian Distributions using
- Witten, D., R. Tibshirani, and T. Hastie (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics 10*(3), 515–534.

Learning a Sparse Coding Dictionary

- Alternatively, we can use a non-parametric prior, such as the beta process

- This allows the model to use dictionaries of unbounded size, rather than having to specify L in advance.

- One can perform Bayesian inference in this model using e.g., Gibbs sampling or variational Bayes.

- One finds that the effective size of the dictionary goes down as the noise level goes up, due to the Bayesian Occam's razor.

- This can prevent overfitting. See (Zhou et al. 2009) for details.

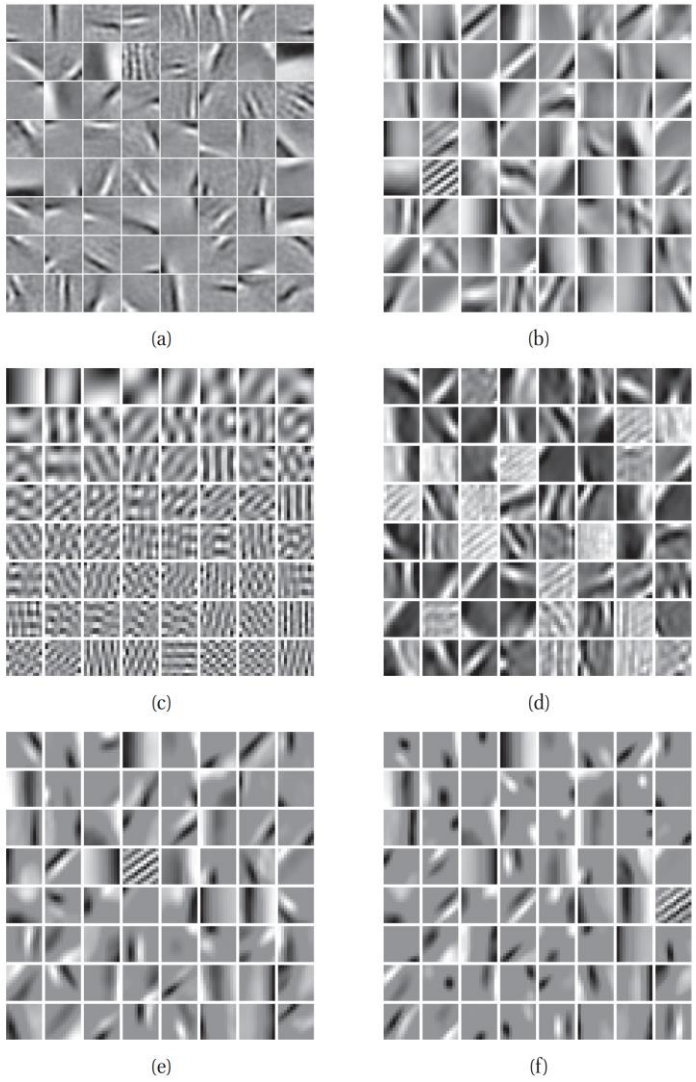
- Zhou, H., D. Karakos, S. Khudanpur, A. Andreou, and C. Priebe (2009). On Projections of Gaussian Distributions using

Results of Dictionary Learning from Image Patches

- One reason that sparse coding has generated so much interest recently is because it explains an interesting phenomenon in neuroscience.
- In particular, the dictionary that is learned by applying sparse coding to patches of natural images consists of basis vectors that look like the filters that are found in simple cells in the primary visual cortex of the mammalian brain (Olshausen and Field 1996).
- In particular, the filters look like bar and edge detectors.
- Interestingly, using ICA gives visually similar results.
- By contrast, applying PCA to the same data results in sinusoidal gratings
  - these do not look like cortical cell response patterns

- Olshausen, B. A. and D. J. Field (1996). Emergence of simple cell receptive field properties by learning a sparse code for natural images. *Nature 381*, 607–609.
- Hyvarinen, A., J. Hurri, and P. Hoyer (2009). *Natural Image Statistics: a probabilistic approach to early computational vision*. Springer.

Results of Dictionary Learning from Image Patches



- Illustration of the filters learned by various methods when applied to natural image patches. (Each patch is first centered and normalized to unit norm.)
- (a) ICA.
- (b) sparse coding.
- (c) PCA.
- (d) non-negative matrix factorization.
- (e) sparse PCA with low sparsity on weight matrix.
- (f) sparse PCA with high sparsity on weight matrix.

icaBasisDemo.m
from PMTK3

sparseDictDemo.m
from PMTK3

Compressed Sensing

- Imagine that, instead of observing the data $\mathbf{x} \in \mathbb{R}^D$, we observe a low-dimensional projection of it, $\mathbf{y} = \mathbf{R}\mathbf{x} + \boldsymbol{\epsilon}$ where $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{R}$ is a $M \times D$ matrix, $M \ll D$ and $\boldsymbol{\epsilon}$ is noise term.
- We assume $\mathbf{R}$ is known sensing matrix, corresponding to different linear projections of $\mathbf{x}$.
- Our goal is to infer $p(\mathbf{x}|\mathbf{y}, \mathbf{R})$.
- We use Bayesian inference with an appropriate prior, that exploits the fact that natural signals can be expressed as a weighted combination of a small number of suitably chosen basis functions
  - That is we assume $\mathbf{x} = \mathbf{W}\mathbf{z}$ where $\mathbf{z}$ has sparse priors and $\mathbf{W}$ is a suitable dictionary.
- This is called compressed sensing or compressive sensing (Candes et al. 2006; Baruniak 2007)

- Baruniak, R. (2007). Compressive sensing. *IEEE Signal Processing Magazine*.
- Candes, E. and M. Wakin (2008,March). An introduction to compressive sampling. *IEEE Signal Processing Magazine 21*.
- Candes, E., M. Wakin, and S. Boyd (2008). Enhancing sparsity by reweighted l1 minimization. *J. of Fourier Analysis and Applications 1*, 877–905.

Compressed Sensing

- For CS to work, it is important to represent the signal in the right basis, otherwise it will not be sparse.
- In traditional CS applications, the dictionary is fixed to be a standard form, such as wavelets.
- However, one can get much better performance by learning a domain-specific dictionary using sparse coding (Zhou et al. 2009).
- As for the sensing matrix $\boldsymbol{R}$, it is often chosen to be a random matrix, for reasons explained in (Candes and Wakin 2008).
- However, one can get better performance by adapting the projection matrix to the dictionary (Seeger and Nickish 2008; Chang et al. 2009)

- Zhou, M., H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin (2009). Non-parametric Bayesian Dictionary Learning for Sparse Image Representations. In *NIPS*.
- Baruniak, R. (2007). Compressive sensing. *IEEE Signal Processing Magazine*.
- Candes, E. and M. Wakin (2008,March). An introduction to compressive sampling. *IEEE Signal Processing Magazine 21*.
- Candes, E., M. Wakin, and S. Boyd (2008). Enhancing sparsity by reweighted l1 minimization. *J. of Fourier Analysis and Applications 1*, 877–905.
- Seeger, M. and H. Nickish (2008). Compressed sensing and Bayesian experimental design. In *Intl. Conf. on Machine Learning*.
- Chang, J., J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei (2009). Reading tea leaves: How humans interpret topic models. In *NIPS*.

Image inpainting and denoising

- Suppose we have an image which is corrupted in some way, e.g., by having text or scratches sparsely superimposed on top of it.
- We might want to estimate the underlying "clean" image.
- This is called image inpainting.
- One can use similar techniques for image denoising.
- We can model this as a special kind of compressed sensing problem.
- We partition the image into overlapping patches, $\boldsymbol{y}_i$ and concatenate them to form $\boldsymbol{y}$.
- We define $\mathbf{R}$ so that the i-th row selects out patch i.
- Now define $\mathcal{V}$ to be the visible (uncorrupted) components of $\boldsymbol{y}$ and $\mathcal{H}$ to be the hidden components.
- To perform image inpainting, we just compute $p(\boldsymbol{y}_{\mathcal{H}}|\boldsymbol{y}_{\mathcal{V}}, \boldsymbol{\theta})$.

Image Inpainting and Denoising

- $\boldsymbol{\theta}$ represents model parameters which specify the dictionary $\mathbf{W}$ and sparsity level λ of $\boldsymbol{z}$
- We can either learn a dictionary offline from a database of images, or we can learn a dictionary just for this image, based on the non-corrupted patches

(a) (b)

An example of image inpainting using sparse coding. Left: original image. Right: reconstruction.

Image Inpainting and Denoising

- The dictionary (of size 256 atoms) was learned from 7×10^6 undamaged 12×12 color patches in the 12 mega-pixel image.

- An alternative approach is to use a graphical model (e.g., the fields of experts model (S. Roth and M. Black, 2009)) which directly encodes correlations between neighboring image patches, rather than using a latent variable model.

  - Unfortunately such models tend to be computationally more expensive.

- S. Roth and M. Black (2009, April). Fields of experts. *Intl. J. Computer Vision 82*(2), 205–229.